

PENGEMBANGAN APLIKASI *MOBILE TRACER STUDY* MENGUNAKAN *PLATFORM* ANDROID

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Ahmad Kamil Almasyhur

NIM: 145150200111170



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PENGEMBANGAN APLIKASI *MOBILE TRACER STUDY*
MENGUNAKAN *PLATFORM ANDROID*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

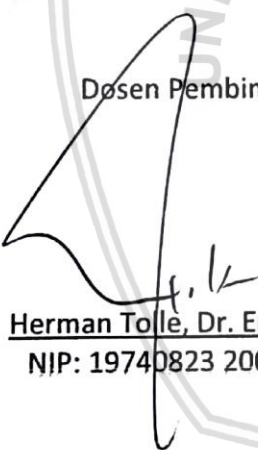
Disusun Oleh :
Ahmad Kamil Almasyhur
NIM: 145150200111170

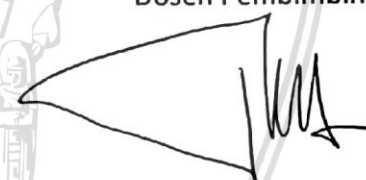
Skrripsi ini telah diuji dan dinyatakan lulus pada
6 Juli 2018

Telah diperiksa dan disetujui oleh:



Dosen Pembimbing 1

Dosen Pembimbing 2


Herman Tolle, Dr. Eng., S.T, M.T
NIP: 19740823 200012 1 001


Adam Hendra Brata, S.Kom., M.T., M.Sc.
NIK: 201607 900105 1 001

Mengetahui
Ketua Jurusan Teknik Informatika



Pri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 15 Mei 2018



Ahmad Kamil Almasyhur

NIM: 145150200111170

KATA PENGANTAR

Puji syukur penulis panjatkan atas kehadiran Tuhan yang Maha Esa, karena telah melimpahkan berkat, rahmat, dan anugerah-Nya kepada penulis, sehingga penulis dapat menyelesaikan skripsi yang berjudul **“PENGEMBANGAN APLIKASI MOBILE TRACER STUDY MENGGUNAKAN PLATFROM ANDROID”**. Berkat bimbingan dan dorongan dari pihak-pihak yang telah membantu penulis dalam menyusun skripsi ini, penulis dapat menyelesaikan skripsi dengan lebih baik. Maka dari itu penulis ingin mengucapkan terima kasih kepada pihak-pihak yang telah membantu penulis secara langsung dan tidak langsung. Adapun pihak-pihak yang membantu penulis dalam menyelesaikan skripsi ini antara lain:

1. Allah SWT atas waktu, kesehatan, serta segala nikmat yang diberikan.
2. Ayah (Moch. Sanusi), Ibu (Suud Bin Ali), serta seluruh keluarga besar yang selalu memberikan semangat, nasehat, perhatian, kesabaran dalam mendidik penulis, serta memberikan doa demi terselesaikannya skripsi ini.
3. Bapak Herman Tolle, Dr. Eng., S.T, M.T selaku dosen pembimbing 1, dan Bapak Adam Hendra Brata, S.Kom., M.T., M.Sc selaku dosen pembimbing 2 yang telah membimbing, memberi saran serta motivasi kepada penulis selama penyusunan laporan skripsi.
4. Bapak Hermawan Dwi Putra, S.H, Bapak Prasetyo Iskandar, S.T, Bapak Andito Aryo D.P, S.Psi, Bapak Dhika Wirya Dipraja, S.T, Ibu Retno Indah Rokhmawati, S.Pd., M.Pd, atas bimbingan dan bantuannya dalam proses menyelesaikan skripsi ini.
5. Seluruh Dosen Fakultas Ilmu Komputer, Universitas Brawijaya atas kesediaannya dalam mengajarkan dan membagikan ilmu yang bermanfaat bagi penulis.
6. Teman-teman angkatan 2014, kakak tingkat 2013, praktikan penulis angkatan 2015, 2016, serta 2017 yang tiada henti bertanya tentang progress skripsi penulis.
7. Teman-teman Private Class L yang banyak membantu penulis dari awal kuliah hingga penyelesaian skripsi ini, khususnya dari RPL yang berfaedah.
8. Teman-teman Tahu Telor Community (TTC), dan Basic Computing Community (BCC) yang banyak membantu dalam pengembangan bakat penulis.
9. I Gede Tika Permana yang membantu dalam proses pembuatan logo *Tracer Study*.
10. Seluruh pihak yang telah membantu kelancaran penulisan tugas akhir yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa penulisan skripsi ini jauh dari kata sempurna. Maka untuk itu penulis mengharapkan saran, kritik, serta masukan dari semua pihak, demi kesempurnaan dan perbaikan skripsi ini. Akhir kata semoga skripsi ini dapat memberikan manfaat bagi semua pihak.

Malang, 5 Juni 2018

Ahmad Kamil Almasyhur

ahmadkamilalmasyhur@gmail.com



ABSTRAK

Ahmad Kamil Almasyhur, Pengembangan aplikasi *mobile tracer study* menggunakan *platform* Android

Dosen Pembimbing:

1. Herman Tolle, Dr. Eng., S.T, M.T
2. Adam Hendra Brata, S.Kom., M.T., M.Sc.

Sistem *Tracer Study* adalah sistem yang menghubungkan antara mahasiswa yang sudah lulus dari suatu instansi pendidikan dengan alumni yang lain sehingga memiliki hubungan antara satu alumni dan alumni yang lain. Alumni dapat saling memberikan informasi tentang lowongan pekerjaan kepada alumni yang lain. Alumni juga dapat memberikan informasi pekerjaan alumni saat ini, serta alumni dapat melihat informasi pekerjaan alumni juga. Alumni juga dapat memberikan informasi pendidikan serta melihat informasi pendidikan yang telah diisikan oleh alumni itu sendiri.

Tidak adanya aplikasi yang membantu dalam proses *tracer study* yang dilakukan di Fakultas Ilmu Komputer, Universitas Brawijaya (FILKOM UB), membuat penulis berinisiatif dalam proses pengembangan aplikasi *mobile tracer study* dengan menggunakan *platform* android.

Penelitian ini menggunakan SDLC (*Software Development Lifecycle*) *prototype* dimana yang dituliskan pada naskah skripsi ini adalah *prototype* terakhir, atau *prototype* final. Dengan menggunakan *platform* firebase dan android, diharapkan sistem *tracer study* mampu membantu dalam proses *tracer study* yang dilakukan di FILKOM UB, dengan hasil pengujian usabilitas 75, dalam rentang 0 – 100 termasuk kedalam kualifikasi yang cukup baik, kemudian hasil pengujian fungsionalitas dan kompatibilitas yang dihasilkan oleh aplikasi *tracer study* mampu mencapai nilai sempurna pada sistem operasi *nougat* dan *marshmallow*.

Kata kunci: *tracer study*, firebase, android, *chat*.

ABSTRACT

Ahmad Kamil Almasyhur, Pengembangan aplikasi *mobile tracer study* menggunakan *platform* Android

Supervisors:

1. Herman Tolle, Dr. Eng., S.T, M.T
2. Adam Hendra Brata, S.Kom., M.T., M.Sc.

Tracer study system is a system that connects students who have graduated from educational institution with their friends who have graduated, which call as alumni. Alumni can give each other information about job vacancy to other alumni, the alumni can also provide information on current alumni work. Alumni can view information on alumni's work as well, then alumni can also provide educational information and view educational information that has been filled by the alumni themselves.

There is no application that helps in the process of tracer study conducted in Faculty of Computer Science, Universitas Brawijaya (FILKOM UB), make the author the initiative in the process of developing mobile tracer study application using android platform.

This study uses SDLC (Software Development Lifecycle) of prototype, which is written on the script of this thesis is the last prototype, or the final prototype. By using the firebase and android platform, it is expected that the tracer study system can assist in the process of tracer study in FILKOM UB, with the result of the usability test is 75 (in range 0 – 100) included in the qualification is good enough, the result of functionality and compatibility generated by tracer study application capable of achieving perfect value on nougat and marshmallow operating system.

Keywords: tracer study, firebase, android, chat.

DAFTAR ISI

| | |
|--|------|
| PENGESAHAN | ii |
| PERNYATAAN ORISINALITAS | iii |
| KATA PENGANTAR..... | iv |
| ABSTRAK..... | vi |
| ABSTRACT | vii |
| DAFTAR TABEL..... | xi |
| DAFTAR GAMBAR | xiii |
| DAFTAR LAMPIRAN | xv |
| BAB 1 PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah..... | 4 |
| 1.3 Tujuan | 4 |
| 1.4 Manfaat..... | 4 |
| 1.5 Batasan Masalah | 5 |
| 1.6 Sistematika Pembahasan | 5 |
| BAB 2 LANDASAN KEPUSTAKAAN | 7 |
| 2.1 Kajian Pustaka | 7 |
| 2.2 Android SDK <i>Platform</i> | 7 |
| 2.3 Firebase..... | 8 |
| 2.4 Firebase Realtime Database | 10 |
| 2.5 JavaScript Object Notation (JSON)..... | 11 |
| 2.6 Pengujian | 13 |
| 2.6.1 Pengujian Fungsional | 13 |
| 2.6.2 Pengujian Kompatibilitas | 14 |
| 2.6.3 Pengujian Usabilitas | 14 |
| BAB 3 METODOLOGI PENELITIAN | 15 |
| 3.1 Studi Pustaka..... | 15 |
| 3.2 Rekayasa Kebutuhan..... | 16 |
| 3.2.1 Identifikasi Aktor | 17 |
| 3.2.2 Identifikasi Kebutuhan | 17 |

| | |
|---|----|
| 3.2.3 Kebutuhan Fungsional..... | 17 |
| 3.2.4 Kebutuhan Non-Fungsional | 17 |
| 3.2.5 Pemodelan Kebutuhan..... | 18 |
| 3.2.6 Analisis Pertanyaan <i>Tracer Study</i> | 19 |
| 3.3 Perancangan | 19 |
| 3.3.1 <i>Diagram Activity</i> | 19 |
| 3.3.2 Perancangan Data | 20 |
| 3.3.3 Perancangan Antarmuka..... | 20 |
| 3.3.4 <i>Screenflow Diagram</i> | 20 |
| 3.3.5 Perancangan Tugas Pengguna | 20 |
| 3.4 Implementasi | 20 |
| 3.5 Pengujian | 21 |
| 3.6 Analisis Hasil Pengujian..... | 22 |
| 3.7 Penarikan Kesimpulan dan Saran | 22 |
| BAB 4 REKAYASA KEBUTUHAN..... | 23 |
| 4.1 Gambaran Umum Sistem..... | 23 |
| 4.2 Analisis Kebutuhan | 24 |
| 4.3 Identifikasi Aktor..... | 25 |
| 4.4 Identifikasi Kebutuhan..... | 25 |
| 4.4.1 Kebutuhan Fungsional..... | 26 |
| 4.4.2 Kebutuhan Non Fungsional..... | 27 |
| 4.5 Pemodelan Kebutuhan | 28 |
| 4.5.1 <i>Diagram Use Case</i> | 28 |
| 4.5.2 <i>Use Case Scenario</i> | 29 |
| 4.6 Analisis Data..... | 35 |
| 4.7 Analisis <i>User Experience</i> | 35 |
| 4.8 Analisis Pertanyaan <i>Tracer Study</i> | 36 |
| BAB 5 PERANCANGAN..... | 38 |
| 5.1 Perancangan Arsitektur Sistem..... | 38 |
| 5.2 <i>Diagram Activity</i> | 39 |
| 5.2.1 Masukkan Informasi Pekerjaan..... | 39 |
| 5.2.2 Masukkan Informasi Pendidikan..... | 40 |

| | |
|--|-----|
| 5.2.3 Melihat Data Pekerjaan..... | 41 |
| 5.3 Perancangan Data..... | 41 |
| 5.4 Perancangan Antarmuka | 54 |
| 5.4.1 <i>Screenflow Diagram</i> | 54 |
| 5.4.2 Perancangan Antarmuka..... | 55 |
| 5.5 Perancangan Tugas Pengguna | 61 |
| BAB 6 IMPLEMENTASI | 64 |
| 6.1 Implementasi | 64 |
| 6.1.1 Implementasi Kode Program | 64 |
| 6.1.2 Implementasi Data | 75 |
| 6.1.3 Implementasi Antarmuka | 80 |
| BAB 7 PENGUJIAN | 88 |
| 7.1 Pengujian | 88 |
| 7.1.1 Pengujian Fungsional | 90 |
| 7.1.2 Pengujian Kompatibilitas | 98 |
| 7.1.3 Pengujian Usabilitas | 99 |
| 7.2 Hasil Pengujian..... | 102 |
| 7.2.1 Hasil Pengujian Fungsional..... | 102 |
| 7.2.2 Hasil Pengujian Kompatibilitas..... | 104 |
| 7.2.3 Hasil Pengujian Usabilitas | 109 |
| 7.3 Analisis Hasil Pengujian..... | 111 |
| 7.3.1 Analisis Hasil Pengujian Fungsional | 111 |
| 7.3.2 Analisis Hasil Pengujian Kompatibilitas..... | 112 |
| 7.3.3 Analisis Hasil Pengujian Usabilitas | 112 |
| BAB 8 PENUTUP | 114 |
| 8.1 Kesimpulan..... | 114 |
| 8.2 Saran | 114 |
| Daftar pustaka..... | 116 |
| LAMPIRAN A HASIL WAWANCARA..... | 118 |
| LAMPIRAN B FORM PERTANYAAN TRACER STUDY | 121 |
| LAMPIRAN C FORM YANG TELAH DIISI OLEH ALUMNI | 123 |
| LAMPIRAN D HASIL PENGUJIAN USABILITAS (SUS)..... | 130 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 4.1 Daftar Aktor | 25 |
| Tabel 4.2 Daftar Kebutuhan Fungsional <i>Guest</i> | 26 |
| Tabel 4.3 Daftar Kebutuhan Fungsional Alumni | 27 |
| Tabel 4.4 Kebutuhan Non Fungsional | 28 |
| Tabel 4.5 Skenario <i>Login</i> (TS-F-G-101) | 30 |
| Tabel 4.6 Skenario <i>Register</i> (TS-F-G-201) | 30 |
| Tabel 4.7 Skenario Kirim Pesan (TS-F-AL-101) | 31 |
| Tabel 4.8 Skenario Melihat Lowongan Pekerjaan (TS-F-AL-201) | 31 |
| Tabel 4.9 Skenario Ubah Informasi Diri Alumni (TS-F-AL-301) | 32 |
| Tabel 4.10 Skenario Memasukkan Informasi Pekerjaan (TS-F-AL-401) | 32 |
| Tabel 4.11 Skenario Memasukkan Informasi Pendidikan (TS-F-AL-501) | 33 |
| Tabel 4.12 Skenario Memasukkan Lowongan Kerja (TS-F-AL-601) | 33 |
| Tabel 4.13 Skenario Melihat Data Pendidikan (TS-F-AL-701) | 34 |
| Tabel 4.14 Skenario Melihat Data Pekerjaan (TS-F-AL-801) | 34 |
| Tabel 4.15 Tugas Pengguna | 36 |
| Tabel 5.1 Perancangan Data | 41 |
| Tabel 5.2 Informasi <i>Screenflow Tracer Study</i> | 54 |
| Tabel 5.3 Informasi Elemen Halaman <i>Register</i> | 56 |
| Tabel 5.4 Informasi Elemen Halaman <i>Login</i> | 57 |
| Tabel 5.5 Informasi Elemen Halaman <i>Chat</i> | 58 |
| Tabel 5.6 Informasi Elemen Halaman Lowongan Pekerjaan | 59 |
| Tabel 5.7 Informasi Elemen Halaman Profil | 60 |
| Tabel 6.1 Kode Masukkan Informasi Pekerjaan | 64 |
| Tabel 6.2 Kode Masukkan Informasi Pendidikan | 69 |
| Tabel 6.3 Kode Melihat Data Pekerjaan | 73 |
| Tabel 7.1 Spesifikasi Perangkat Keras Android Samsung Galaxy S6 | 88 |
| Tabel 7.2 Spesifikasi Perangkat Keras Android Samsung J2 Pro (2016) | 89 |
| Tabel 7.3 Spesifikasi Perangkat Keras Android Emulator | 89 |
| Tabel 7.4 Tabel Sistem Operasi Perangkat Keras | 90 |
| Tabel 7.5 Kasus Uji <i>Login</i> | 90 |

| | |
|---|-----|
| Tabel 7.6 Kasus Uji <i>Login</i> | 91 |
| Tabel 7.7 Kasus Uji <i>Login</i> | 91 |
| Tabel 7.8 Kasus Uji <i>Login</i> | 92 |
| Tabel 7.9 Kasus Uji <i>Login</i> | 92 |
| Tabel 7.10 Kasus Uji <i>Register</i> | 93 |
| Tabel 7.11 Kasus Uji <i>Register</i> | 93 |
| Tabel 7.12 Kasus Uji Kirim Pesan | 94 |
| Tabel 7.13 Kasus Uji Melihat Lowongan Pekerjaan | 94 |
| Tabel 7.14 Kasus Uji Ubah Informasi Diri | 95 |
| Tabel 7.15 Kasus Uji Masukkan Informasi Pekerjaan | 95 |
| Tabel 7.16 Kasus Uji Masukkan Informasi Pekerjaan | 96 |
| Tabel 7.17 Kasus Uji Masukkan Lowongan Pekerjaan | 96 |
| Tabel 7.18 Kasus Uji Melihat Informasi Pendidikan | 97 |
| Tabel 7.19 Kasus Uji Melihat Informasi Pekerjaan | 97 |
| Tabel 7.20 Kasus Uji Kompatibilitas | 98 |
| Tabel 7.21 Kasus Uji Tingkat Keberhasilan Pengguna | 100 |
| Tabel 7.22 Kuesioner Pengujian Usabilitas | 101 |
| Tabel 7.23 Hasil Pengujian Fungsional | 102 |
| Tabel 7.24 Hasil Pengujian Kompatibilitas (Samsung Galaxy S6) | 104 |
| Tabel 7.25 Hasil Pengujian Kompatibilitas (Samsung J2 Pro 2016) | 106 |
| Tabel 7.26 Hasil Pengujian Kompatibilitas (Emulator) | 107 |
| Tabel 7.27 Hasil Pengujian Usabilitas | 109 |
| Tabel 7.28 Hasil Pengujian Usabilitas | 110 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2.1 Struktur Firebase | 8 |
| Gambar 2.2 JSON <i>Object</i> | 12 |
| Gambar 2.3 JSON Array | 12 |
| Gambar 2.4 JSON Value | 13 |
| Gambar 3.1 Diagram Alir Metodologi | 15 |
| Gambar 3.2 Gambaran Umum Sistem | 18 |
| Gambar 4.1 <i>Story Board Tracer Study</i> | 24 |
| Gambar 4.2 Aturan Penomoran Kebutuhan | 26 |
| Gambar 4.3 <i>Use Case Diagram</i> | 29 |
| Gambar 3.2 Gambaran Umum Sistem | 38 |
| Gambar 5.1 Masukkan Informasi Pekerjaan | 39 |
| Gambar 5.2 Memasukkan Informasi Pendidikan | 40 |
| Gambar 5.3 Melihat Data Pekerjaan | 41 |
| Gambar 5.4 Screenflow <i>Tracer Study</i> | 55 |
| Gambar 5.5 Halaman <i>Register</i> | 56 |
| Gambar 5.6 Halaman <i>Login</i> | 57 |
| Gambar 5.7 Halaman <i>Chat</i> | 58 |
| Gambar 5.8 Halaman Lowongan Pekerjaan | 59 |
| Gambar 5.9 Halaman Profil | 60 |
| Gambar 6.1 Implementasi Data <i>List Chat</i> Alumni | 75 |
| Gambar 6.2 Implementasi Data <i>Chat</i> Alumni | 76 |
| Gambar 6.3 Implementasi Data User Alumni | 77 |
| Gambar 6.4 Implementasi Data Pekerjaan Alumni | 77 |
| Gambar 6.5 Implementasi Data User Alumni | 78 |
| Gambar 6.6 Implementasi Data Pendidikan Alumni | 78 |
| Gambar 6.7 Implementasi Data Timeline untuk Alumni | 79 |
| Gambar 6.8 Antarmuka Halaman <i>Chat</i> | 80 |
| Gambar 6.9 Antarmuka Halaman Ruang <i>Chat</i> | 81 |
| Gambar 6.10 Antarmuka Halaman Lowongan Pekerjaan | 82 |
| Gambar 6.11 Antarmuka Halaman Profil | 83 |

| | |
|---|----|
| Gambar 6.12 Antarmuka Halaman Tambah Riwayat Pendidikan | 84 |
| Gambar 6.13 Antarmuka Halaman Tambah Riwayat Pekerjaan | 85 |
| Gambar 6.14 Antarmuka Halaman Lihat Riwayat Pendidikan..... | 86 |
| Gambar 6.15 Antarmuka Halaman Lihat Riwayat Pekerjaan..... | 87 |
| Gambar 7.1 Aturan Penomoran Pengujian Fungsional | 90 |
| Gambar 7.2 Aturan Penomoran Pengujian Kompatibilitas..... | 98 |



DAFTAR LAMPIRAN

| | |
|--|-----|
| Lampiran 1 Kemahasiswaan Fakultas Ilmu Komputer | 118 |
| Lampiran 2 Alumni Fakultas Ilmu Komputer | 120 |
| Lampiran 3 Pertanyaan Tracer Study | 121 |
| Lampiran 4 Data Alumni FILKOM (Fathony Teguh Irawan) | 123 |
| Lampiran 5 Data Alumni FILKOM (Novi Nur Putriwijaya) | 125 |
| Lampiran 6 Data Alumni FILKOM (Muhammad Ridho KP) | 126 |
| Lampiran 7 Data Alumni FILKOM (Hilmi Ilyas) | 128 |
| Lampiran 8 Hasil Pengujian Usabilitas | 130 |



DAFTAR LAMPIRAN

| | |
|--|-----|
| Lampiran 1 Kemahasiswaan Fakultas Ilmu Komputer | 118 |
| Lampiran 2 Alumni Fakultas Ilmu Komputer | 120 |
| Lampiran 3 Pertanyaan Tracer Study | 121 |
| Lampiran 4 Data Alumni FILKOM (Fathony Teguh Irawan) | 123 |
| Lampiran 5 Data Alumni FILKOM (Novi Nur Putriwijaya) | 125 |
| Lampiran 6 Data Alumni FILKOM (Muhammad Ridho KP) | 126 |
| Lampiran 7 Data Alumni FILKOM (Hilmi Ilyas) | 128 |
| Lampiran 8 Hasil Pengujian Usabilitas | 130 |



BAB 1 PENDAHULUAN

Bagian utama skripsi terdiri dari beberapa komponen atau bab yang tersusun dengan alur yang logis. Pendahuluan merupakan komponen/bab pertama yang harus menjelaskan apa yang dikerjakan dalam skripsi dan mengapa ini dikerjakan.

1.1 Latar Belakang

Tracer Study adalah sebuah layanan yang disediakan oleh setiap lembaga pendidikan untuk melacak alumni yang telah lulus dari lembaganya yang mana nantinya dapat digunakan untuk akreditasi dan pendataan para lulusan yang ada. Pelacakan alumni adalah salah satu studi empiris yang diharapkan mampu menyediakan informasi hasil pendidikan di sebuah lembaga universitas. Informasi yang didapat kemudian dapat digunakan untuk pengembangan lebih lanjut dalam menjamin kualitas pendidikan. Dengan kegiatan pelacakan alumni ini, Universitas Brawijaya (UB) dapat memperoleh informasi dan umpan balik yang berkaitan dengan kekurangan pelaksanaan program studi dan penyediaan dasar-dasar pelaksanaan perencanaan di masa depan. Untuk itu informasi keberhasilan profesionalisme (karier, status, pendapatan) para alumni dibutuhkan. Demikian pula informasi terhadap pengetahuan dan keahlian yang relevan (hubungan antara pengetahuan dan keahlian dengan kebutuhan kerja, ruang lingkup pekerjaan, posisi profesional). Para alumni diharapkan dapat memberikan evaluasi terhadap sistem pembelajaran dan kondisi yang mereka alami ketika mereka masih belajar di UB kemudian dikaitkan dengan dunia kerja yang mereka hadapi. (Job Placement Center dan *Tracer Study* Universitas Brawijaya, 2018)

Dikti saat ini sedang melaksanakan upaya untuk merintis kompilasi data *tracer study* nasional khususnya mengenai transisi dan posisi pekerjaan lulusan PT di Indonesia. Untuk itu sejak tahun 2011, telah dikembangkan suatu sistem online yang dapat digunakan oleh Perguruan Tinggi untuk melacak aktivitas para lulusannya setelah masa pendidikan tinggi, baik masa transisi maupun pergerakan mereka di dunia kerja sampai. *tracer study* dinilai penting karena menjadi alat evaluasi kinerja PT dan sekarang telah dijadikan salah satu syarat kelengkapan akreditasi oleh Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT), sebagai kelengkapan dalam dokumen Evaluasi Diri yang diperlukan dalam pengajuan proposal melalui Kemdikbud.

Tracer study Dikti ditujukan untuk melacak jejak lulusan/alumni yang dilakukan 2 tahun setelah lulus dan bertujuan untuk mengetahui beberapa hal, yang pertama adalah *outcome* pendidikan dalam bentuk transisi dari dunia pendidikan tinggi ke dunia kerja (termasuk masa tunggu kerja dan proses pencarian kerja pertama), situasi kerja terakhir, dan aplikasi kompetensi di dunia kerja. Yang kedua adalah *output* pendidikan yaitu penilaian diri terhadap penguasaan dan pemerolehan kompetensi. Dan yang ketiga adalah proses pendidikan berupa evaluasi proses pembelajaran dan kontribusi pendidikan

tinggi terhadap pemerolehan kompetensi. Yang mana nantinya dari ketiga tujuan tersebut diharapkan dapat membantu proses terhubungnya hubungan antara alumni dengan alumni yang lain, alumni dengan instansi pendidikannya, serta alumni dengan adik tingkat alumni.

Hasil *tracer study* akan membantu Perguruan Tinggi dalam mengetahui posisi lulusan yang telah terserap dalam dunia kerja serta menyiapkan lulusan sesuai dengan kompetensi yang diperlukan di dunia kerja. Hasil *tracer study* yang kemudian dilaporkan ke Dikti akan membantu program Pemerintah dalam rangka memetakan kebutuhan dunia kerja dengan pembangunan pendidikan di Indonesia. (DIKTI, n.d.)

Tracer Study merupakan kegiatan sensus, untuk kemudian dianalisis sebagai salah satu cara dalam pengembangan institusi, penyusunan kurikulum perkuliahan, mendukung kegiatan akreditasi dan juga membangun jaringan alumni. Dari *tracer study* ini, pihak universitas dapat melakukan penyelarasan pendidikan dengan dunia kerja, serta mengetahui hasil, pengeluaran, dan proses pendidikan yang sudah berlangsung. Menurut Dikti, seluruh perguruan tinggi di Indonesia wajib untuk melakukan kegiatan ini. Namun, pada praktiknya baru 40% perguruan tinggi yang sudah menjalankan program *tracer study* (Mfc, 2015)

Tracer study sangat dibutuhkan dalam lingkup pendidikan sekarang ini karena akan memetakan kebutuhan dunia kerja, memetakan persebaran lulusan / alumni, dan akan membantu banyak pihak dalam proses pendidikan. Sedangkan banyak Instansi pendidikan yang belum memiliki aplikasi yang mendukung jalannya proses *tracer study*. Oleh karena *tracer study* sangat dibutuhkan dalam dunia pendidikan karena dalam proses perkembangan menyelaraskan kebutuhan pekerjaan dengan pendidikan maka dibutuhkan sebuah aplikasi *mobile tracer study* untuk melacak lulusan. Jika sebuah instansi pendidikan tidak memiliki sebuah aplikasi yang mendukung proses untuk melacak lulusan maka akan sulit untuk mendapatkan predikat sangat baik (A) dalam penilaian yang dilakukan oleh dikti.

Pengguna aplikasi *tracer study* sudah dipastikan adalah lulusan / alumni dari instansi pendidikan yang ada. Dan pengguna aplikasi *tracer study* adalah pekerja / wiraswasta yang mana pengguna internet terbanyak berdasarkan hasil survey yang dilakukan oleh Asosiasi Penyedia Jasa Internet Indonesia (APJII) pengguna internet Indonesia dinominasi oleh para pekerja / wiraswasta dengan porsi 62% dari total pengguna internet. Sedangkan pengguna internet berdasarkan usia, pengguna internet terbesar berumur 25 – 44 yang mana usia produktif untuk bekerja / berwiraswasta. Pengguna internet berdasarkan usia, pengguna terbanyak adalah usia 35-44 tahun sebesar 29,2%. Sedangkan data berdasarkan pekerjaan, pengguna internet terbanyak berprofesi sebagai Pekerja / Wiraswasta sebesar 82,2 juta atau 62% (Isparmo, 2016). Dengan perbedaan umur dari pengguna internet, dibutuhkan sebuah aplikasi yang memiliki tingkat kegunaan yang baik, dimana mendukung baik untuk pengguna yang berumur 25 hingga untuk pengguna yang berumur 44 tahun.

Laporan terbaru dari perusahaan riset pasar comScore menunjukkan setidaknya ada 63,6 juta pengguna internet atau kerap disebut "netizen" di Tanah Air. Angka itu dihimpun sepanjang Januari 2017. Dari total tersebut, sebanyak 21,2 juta orang masih mengakses internet dari desktop. Sementara itu, mayoritas sudah mengakses internet via perangkat *mobile* seperti smartphone dan tablet, yakni sebanyak 51,7 juta orang. Dari 51,7 juta pengguna internet *mobile*, sekitar 46 juta merupakan pengguna aplikasi *mobile* dan 40 juta adalah pengguna situs *mobile* (Kompas, 2017). Jumlah terbanyak dari pengguna internet adalah pengguna aplikasi *mobile* yaitu sejumlah 46 juta, sedangkan selanjutnya adalah pengguna situs *mobile* dengan jumlah 40 juta orang. dari total 46 juta pengguna aplikasi *mobile* di Indonesia, Android adalah yang terbanyak digunakan.

Android, sistem operasi besutan Google, makin menunjukkan kedigdayaannya di Asia Tenggara. Di seluruh negara Asia Tenggara yang tentunya termasuk Indonesia, Android menguasai lebih dari separuh pangsa pasar. Data tersebut tertuang dalam laporan kuartal II yang disusun oleh biro marketing bernama Waiwai Marketing. Mereka mengumpulkan data dari facebook untuk memperkirakan jumlah ponsel yang digunakan di Asia Tenggara.

Indonesia tercatat sebagai negara di Asia Tenggara yang warganya terbanyak menggunakan Android. Totalnya yakni pengguna 41 juta pengguna atau pangsa pasarnya 94%. Sementara iOS di Indonesia hanya digunakan 2,8 juta pengguna atau 6%. Sementara Singapura menjadi negara dengan penetrasi yang cukup berimbang antara Android dan iOS. Di sana, Android digunakan sebanyak 2,1 juta pengguna atau 58%, sedangkan iOS sebanyak 1,5 juta pengguna atau 42%. (Rachman, 2015). Karena banyaknya jenis Android yang ada, dimulai dari Android 1.0 yang dikeluarkan oleh google ditahun 2008, hingga Android 7.0 yang dikeluarkan oleh google ditahun 2016, maka terdapat perbedaan dalam setiap pengembangan Android yang ada, dibutuhkan kompatibilitas yang baik dalam pengembangan aplikasi yang menggunakan Android. Pengguna Android Jelly Bean atau lebih dikenal dengan API 16 sudah mencapai 99.2%, serta dukungan yang tersedia dalam pengembangan aplikasi *tracer study* ini adalah minimal API 16. (Bohang, 2016)

Banyaknya jenis Android yang ada saat ini, maka tinggal kebergunaan aplikasi menjadi terbatas dengan kesesuaian pembuatan aplikasi dengan perangkat keras dan perangkat lunak (sistem operasi yang dibangun). Dengan banyaknya perbedaan perangkat keras dan perangkat lunak, maka tinggal usabilitas dari aplikasi akan menjadi berkurang. Jika tingkat usabilitas dari suatu aplikasi berkurang maka berkurang pula kualitas dari aplikasi tersebut. Namun jika tingkat usabilitas meningkat (fungsionalitas dari aplikasi menjadi mudah digunakan, dan dapat dengan efektif digunakan) maka kualitas dari aplikasi tersebut akan meningkat, seiring dengan meningkatnya usabilitas maka akan membuat kualitas aplikasi menjadi lebih baik. Dengan berkurangnya tingkat usabilitas yang ada pada suatu aplikasi maka tingkat kepuasan pengguna akan berkurang, sehingga pengguna tidak akan menggunakan aplikasi lebih lanjut.

Kualitas aplikasi memiliki beberapa bagian, yang salah satunya adalah usabilitas, dimana usabilitas dapat didefinisikan oleh 5 komponen kualitas, yaitu *learnability*, efisiensi, *memorability*, *error*, *satisfaction*. *Learnability* adalah bagian dari seberapa mudah pengguna menyelesaikan tugas dasar pada saat aplikasi pertama digunakan. Efisiensi adalah seberapa cepat mereka dapat melakukan tugas tugasnya dibandingkan dengan melakukannya dengan cara lain. *Memorability* adalah seberapa mudah pengguna bisa menggunakan aplikasi untuk beberapa kali penggunaan. *Error* adalah seberapa banyak pengguna melakukan kesalahan saat menggunakan aplikasi, seberapa parah kesalahan ini berdampak, dan seberapa mudah pengguna menggunakan aplikasi ketika terjadi kesalahan. Sedangkan *satisfaction* adalah tingkat kepuasan pengguna dalam menggunakan aplikasi yang telah dikembangkan. (Nielsen Norman Group, 2018)

Dengan adanya aplikasi *mobile tracer study* berbasis Android yang memiliki tingkat usabilitas yang baik, diharapkan dapat membantu alumni untuk saling terhubung, dan nantinya dapat memudahkan instansi pendidikan dalam melacak jejak lulusan / alumni dari instansi mereka, sehingga dapat menjadi tolak ukur pendidikan dan juga dapat menjadi penilaian diri terhadap penguasaan dan perolehan kompetensi dan memudahkan instansi dalam melengkapi syarat akreditasi yang dilakukan oleh Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT).

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, terdapat beberapa permasalahan yang akan dibahas pada penelitian ini yaitu sebagai berikut :

1. Bagaimana analisis kebutuhan dalam merancang aplikasi *mobile tracer study* berbasis Android?
2. Bagaimana rancangan aplikasi *mobile tracer study* berbasis Android?
3. Bagaimana implementasi penggunaan *platform* firebase pada aplikasi *tracer study* yang dibangun pada sistem operasi Android?
4. Bagaimana tingkat usabilitas dan kompatibilitas pada aplikasi *tracer study* ini?

1.3 Tujuan

Merancang dan membangun aplikasi *tracer study* dengan menggunakan *platform* Android sehingga memiliki tingkat usabilitas dan kompatibilitas yang baik.

1.4 Manfaat

Manfaat yang diharapkan melalui tugas akhir ini adalah menghasilkan aplikasi *tracer study* yang dapat dimanfaatkan sebagai aplikasi yang memudahkan alumni dari suatu instansi pendidikan untuk saling terhubung dan nantinya dapat membantu instansi pendidikan dalam melacak jejak lulusan / alumni dari instansi

mereka, sehingga dapat menjadi tolak ukur pendidikan dan dalam menjadi penilaian diri terhadap perolehan kompetensi dan memudahkan instansi dalam melengkapi syarat akreditasi yang dilakukan oleh BAN-PT.

1.5 Batasan Masalah

Agar permasalahan yang dirumuskan tidak keluar melebihi konteks permasalahan yang ada, maka penelitian ini dibatasi dalam hal :

1. Perancangan dan implementasi aplikasi *tracer study* ini hanya digunakan pada perangkat dengan sistem operasi Android minimal API 16.
2. Perancangan dan implementasi aplikasi *tracer study* ini dikhususkan untuk menunjang proses *tracer study* di Fakultas Ilmu Komputer Universitas Brawijaya.
3. Aplikasi *tracer study* yang dikembangkan hanya berfokus pada sisi alumni.
4. Aplikasi *tracer study* yang dikembangkan hanya berfokus pada sisi *mobile*.

1.6 Sistematika Pembahasan

Untuk memudahkan pemahaman tentang penelitian ini, maka pembahasan yang ada akan dibagi menjadi beberapa bab yang disesuaikan dengan prosedur penulisan penelitian yang akan dibahas, yaitu :

BAB I Pendahuluan

Bab ini menjelaskan beberapa hal yang akan menjadi latar belakang atas dilakukannya penelitian ini, memaparkan rumusan masalah, tujuan, manfaat, Batasan masalah dari penelitian ini, serta menjelaskan sistematika pembahasan penelitian

BAB II Landasan Kepustakaan

Bab ini menjelaskan beberapa hal mengenai kajian pustaka dan dasar teori yang menjadi dasar dalam pengembangan aplikasi *tracer study*. Diantaranya teori mengenai *library* yang akan digunakan dan hal lainnya.

BAB III Metodologi Penelitian

Bab ini menjelaskan tentang metode yang digunakan dalam pengembangan penelitian ini yang di antara lain adalah studi literature, analisis, perancangan, dan implementasi serta pengujian dan pengambilan kesimpulan.

BAB IV Analisis Kebutuhan dan Perancangan

Bab ini menjelaskan tentang analisis dalam perancangan dari sistem yang akan dibangun yaitu aplikasi *tracer study*.

BAB V Implementasi

Bab ini menjelaskan tentang apa saja yang dilakukan saat proses implementasi dari aplikasi *tracer study*.

BAB VI Pengujian

Bab ini menjelaskan tentang proses pengujian dari aplikasi yang telah diimplementasikan yaitu aplikasi *tracer study*.

BAB VII Penutup

Bab ini berisi kesimpulan dan saran yang dihasilkan dari laporan pengemabangan aplikasi *tracer study*.



BAB 2 LANDASAN KEPUSTAKAAN

Bab ini akan membahas mengenai teori-teori yang akan menjadi dasar dari penelitian ini dan berhubungan dengan rancangan dan implementasi aplikasi *tracer study*. Pada bagian ini juga akan membahas tentang teori-teori dasar tentang Firebase dan Fitur Fitur yang akan digunakan pada implementasi aplikasi *tracer study*.

2.1 Kajian Pustaka

Kajian Pustaka pada penelitian ini akan membahas tentang beberapa penelitian yang telah dilakukan sebelumnya. Beberapa penelitian tersebut akan digunakan peneliti untuk mendukung penelitian dalam skripsi ini. Penelitian-penelitian yang telah dilakukan sebelumnya akan dijelaskan satu persatu oleh peneliti.

Referensi pertama yaitu penelitian dengan judul "Implementing Smart Home Using Firebase" (Kumar, et al., 2016). Penelitian ini menjelaskan tentang apa saja keuntungan menggunakan firebase sebagai *database* utama dalam system mereka yaitu sistem rumah cerdas. Dan mengenai kelebihan kelebihan yang ada di firebase *database* melebihi *database* yang sudah ada atau normal *database*.

Referensi kedua yaitu penelitian dengan judul "Pembangunan Modul Penunjang Pembelajaran Di Kelas Untuk Aplikasi Brawijaya Messenger Dengan Platform Firebase" (Aditya Orba Ramadhan, 2017). Penelitian ini menjelaskan tentang bagaimana menggunakan firebase *database* sebagai sistem untuk penunjang pembelajaran dikelas, *Firestore Database* digunakan sebagai media utama dalam pengembangan aplikasi Brawijaya Messenger. Dalam mengembangkan aplikasi *brawijaya messenger*, fitur yang mungkin digunakan dari *firebase* adalah *Firestore Cloud Messaging* (FCM), *Authentication*, *Realtime Database*, dan *Firestore Storage*.

Referensi ketiga yaitu penelitian dengan judul "Pembangunan Aplikasi Brawijaya Messenger dengan menggunakan *platform Firebase* pada Universitas Brawijaya" (Rozak, et al., 2018). Penelitian ini berfokus pada bagaimana menggunakan firebase untuk menyelesaikan masalah tujuan Messenger yang jarang digunakan untuk membantu media pembelajaran. Penulis tersebut menggunakan *platform* firebase untuk membuat messenger yang akan digunakan untuk menunjang proses pembelajaran yang menggunakan studi kasus di Universitas Brawijaya.

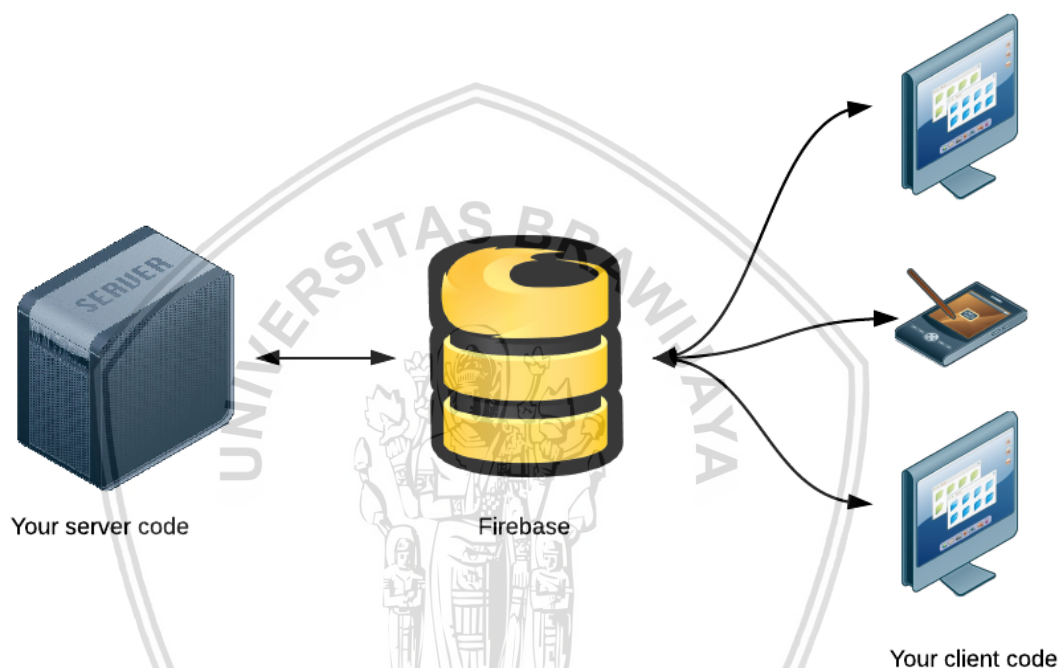
2.2 Android SDK Platform

Saat ini, terdapat 26 jenis SDK (Software Development Kit) yaitu SDK versi 1 hingga SDK versi 26, dimana setiap SDK pasti berisi:

1. Package Android SDK Platform. Ini dibutuhkan untuk mengkompilasi aplikasi Android untuk versi SDK yang dimaksud.

2. Beberapa Package System Image. Paling tidak satu dari System Image (contohnya handsets, Android TV, dan Android Wear) akan digunakan untuk merunning aplikasi di Android Emulator, setiap system image merepresentasikan bagiannya masing masing, (misal untuk Intel x86, dan ARM EABI). System Image yang diberikan label Google APIs dan atau Google Play memiliki Google Play Service.
3. Source dari setiap package Android yang mana Android Studio mungkin menampilkan beberapa baris kode ketika proses debug.

2.3 Firebase



Gambar 2.1 Struktur Firebase

(Sumber: firebase.googleblog.com, 2013)

Firebase adalah *platform* seluler yang membantu developer mengembangkan aplikasi berkualitas tinggi secara cepat, berbasis pengguna, dan dapat menghasilkan uang lebih banyak. Firebase terdiri dari fitur pelengkap yang bisa dipadupadankan sesuai dengan kebutuhan Anda (Google). Digunakannya firebase, karena *platform* yang telah dikembangkan oleh Google ini memiliki fitur cukup mumpuni untuk dimanfaatkan pada aplikasi messenger. Dalam mengembangkan aplikasi brawijaya messenger ini fitur yang mungkin digunakan dari firebase adalah *Firebase Cloud Messaging (FCM)*, *Authentication*, *Realtime Database*, dan *Firebase Storage*. FCM merupakan pengembangan dari Google Cloud Messaging (GCM) yang dapat dimanfaatkan untuk *multiplatform*, beberapa keunggulan dari FCM adalah *upstream/downstream* tanpa batas, mengatasi aspek antrian dan pengiriman. Sedangkan *Authentication* dapat membantu meningkatkan tingkat keamanan permasalahan verifikasi dan validasi

akun, ditambah lagi fitur ini dapat memungkinkan developer untuk mengintegrasikan media terkenal seperti facebook, google, twitter dengan sistem yang dikembangkan. Terakhir *Realtime Database* dan *Storage* adalah fitur yang sangat membantu untuk membuat suatu aplikasi yang bersifat dinamis, karena dengan memanfaatkan fitur ini suatu aplikasi dapat secara dinamis menerima masukan dan membagikannya dengan *realtime* serta menyimpannya kedalam *storage* secara bersamaan. (Aditya Orba Ramadhan, 2017)

Firebase adalah teknologi yang memungkinkan kita membuat aplikasi *web* tanpa pemrograman sisi server sehingga pembangunannya menjadi lebih mudah dan cepat. Dengan menggunakan Firebase, kita tidak perlu menekankan server *over-provisioning* atau membangun *REST API* hanya dengan sedikit konfigurasi. Kita dapat memberi Firebase kesempatan untuk melakukan setiap langkah yang diperlukan: memverifikasi pengguna, menyimpan data, dan menerapkan aturan akses. Ini juga mendukung klien iOS, *web*, OS X, dan Android. Aplikasi yang menggunakan Firebase dapat mengontrol dan menggunakan data, tanpa perlu memikirkan bagaimana data akan disimpan, dan disinkronkan di berbagai contoh aplikasi secara *real time*. Menulis kode sisi server tidak diperlukan saat menggunakan firebase, atau untuk menerapkan kerangka server yang kompleks untuk mendapatkan aplikasi yang dimulai dengan Firebase. (Kumar, et al., 2016)

Kenapa harus menggunakan firebase:

1. *Flexible Backend* dengan kemudahan yang sangat banyak
2. Mengurangi waktu pengembangan perangkat lunak.
3. Data Dikirimkan dan diterima dengan menggunakan JSON, jadi apa yang kita lihat adalah apa yang akan kita terima
4. Data terproteksi dengan aman karena menggunakan enkripsi SSL untuk semua perubahan datanya.
5. Satu konsol dengan banyak produk yang semuanya berfungsi.
6. Didukung oleh Google, dipercaya oleh berbagai aplikasi kelas dunia.

Terdapat 16 Jenis Interaksi di Firebase (Firebase, n.d.)

1. *Realtime Database* : Menyimpan dan menyingkronkan data aplikasi dalam hitungan milidetik.
2. Autentikasi : Autentikasi pengguna dengan mudah dan aman.
3. *Cloud Storage* : Menyimpan dan menayangkan file pada skala Google
4. *Test Lab* : Menguji aplikasi Anda di perangkat yang dihosting oleh Google

5. *Crash Reporting* : Menemukan dan memprioritaskan bug, serta memperbaikinya dengan lebih cepat
6. *Cloud Functions* : Menjalankan kode backend seluler tanpa mengelola server
7. *Hosting* : Kirimkan aset aplikasi *web* dengan cepat dan aman
8. *Pemantauan Performa* : Mendapatkan insight performa aplikasi Anda
9. *Google Analytics* : Mendapatkan analisis aplikasi gratis dan tanpa batas
10. *Dynamic Links* : Mendorong pertumbuhan menggunakan deep link dengan atribusi
11. *Invites* : Mudahkan untuk membagikan aplikasi dan konten Anda
12. *AdMob* : Memaksimalkan pendapatan dengan iklan dalam aplikasi
13. *Remote Config* : Mengubah aplikasi Anda tanpa menerapkan versi baru
14. *App Indexing* : Mendorong traffic penelusuran di aplikasi Anda
15. *AdWords* : Mendorong penginstalan dan mempertahankan pengguna dengan kampanye iklan yang ditargetkan
16. *Cloud Messaging* : Mengirim pesan dan notifikasi yang ditargetkan

2.4 Firebase Realtime Database

Firebase Realtime Database adalah *database* yang di-host di *cloud*. Data disimpan sebagai JSON dan disinkronkan secara *realtime* ke setiap klien yang terhubung. Ketika Kita membuat aplikasi lintas-*platform* dengan SDK Android, iOS, dan JavaScript, semua klien akan berbagi sebuah *instance Realtime Database* dan menerima update data terbaru secara otomatis.

Kelebihan Kelebihan dari *Firebase Realtime Database*:

1. Sebagai ganti permintaan HTTP biasa, *Firebase Realtime Database* menggunakan sinkronisasi data—setiap kali data berubah, semua perangkat yang terhubung akan menerima update dalam waktu milidetik. Memberikan pengalaman yang kolaboratif dan imersif tanpa perlu memikirkan kode jaringan.

2. Aplikasi Firebase tetap responsif bahkan saat *offline* karena SDK Firebase *Realtime Database* menyimpan data ke disk. Setelah konektivitas pulih, perangkat klien akan menerima setiap perubahan yang terlewat dan melakukan sinkronisasi dengan status server saat ini.
3. Firebase *Realtime Database* dapat diakses secara langsung dari perangkat seluler atau browser *web*; server aplikasi tidak diperlukan. Keamanan dan validasi data dapat diakses melalui aturan keamanan Firebase *Realtime Database* yang merupakan kumpulan aturan berbasis ekspresi dan dijalankan ketika data dibaca atau ditulis.
4. Dengan Firebase *Realtime Database* pada paket harga *Blaze*, Anda dapat mendukung kebutuhan data aplikasi Anda pada skala tertentu dengan membagi data Anda di beberapa *instance database* di project Firebase yang sama. Menyederhanakan autentikasi dengan Firebase *Authentication* pada project Anda dan mengautentikasi pengguna di *instance database* Anda. Mengontrol akses ke data di tiap *database* dengan Aturan Firebase *Realtime Database* khusus untuk tiap *instance database*.

Firebase *Realtime Database* memungkinkan Anda untuk membuat aplikasi kolaboratif dan kaya fitur dengan menyediakan akses yang aman ke database, langsung dari kode sisi klien. Data disimpan di drive lokal. Bahkan saat *offline* sekalipun, peristiwa *realtime* terus berlangsung, sehingga pengguna akhir akan merasakan pengalaman yang responsif. Ketika koneksi perangkat pulih kembali, *Realtime Database* akan menyinkronkan perubahan data lokal dengan update jarak jauh yang terjadi selama klien *offline*, sehingga setiap perbedaan akan otomatis digabungkan.

Realtime Database menyediakan bahasa aturan berbasis ekspresi yang fleksibel, atau disebut juga Aturan Keamanan Firebase *Realtime Database*, untuk menentukan metode strukturisasi data dan kapan data dapat dibaca atau ditulis. Ketika diintegrasikan dengan Firebase *Authentication*, developer dapat menentukan siapa yang memiliki akses ke data tertentu dan bagaimana mereka dapat mengaksesnya.

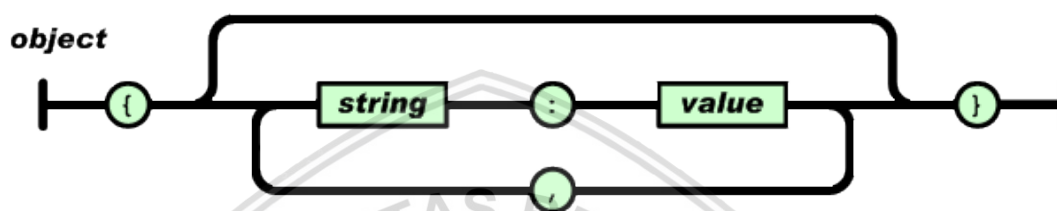
Realtime Database adalah *database* NoSQL, sehingga memiliki pengoptimalan dan fungsionalitas yang berbeda dengan *database* terkait. API *Realtime Database* dirancang agar hanya mengizinkan operasi yang dapat dijalankan dengan cepat. Hal ini memungkinkan Anda untuk membangun pengalaman *realtime* yang luar biasa dan dapat melayani jutaan pengguna tanpa mengorbankan kemampuan respons. Oleh karena itu, perlu dipikirkan bagaimana pengguna mengakses data, kemudian buat struktur data sesuai dengan kebutuhan tersebut.

2.5 JavaScript Object Notation (JSON)

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat

(*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data. (Json, n.d.)

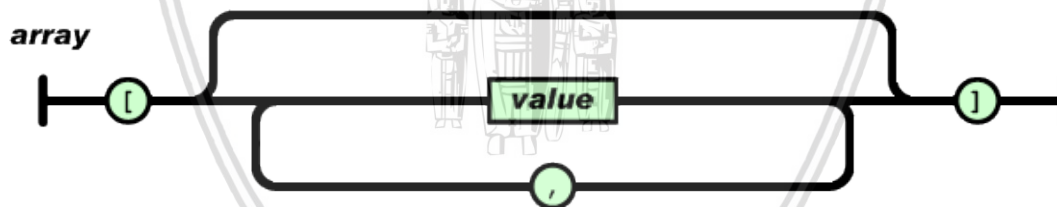
Contoh bentuk json jika data yang dikirimkan adalah *object* adalah sebagai berikut:



Gambar 2.2 JSON Object

(Sumber: <https://www.json.org/>, 2018)

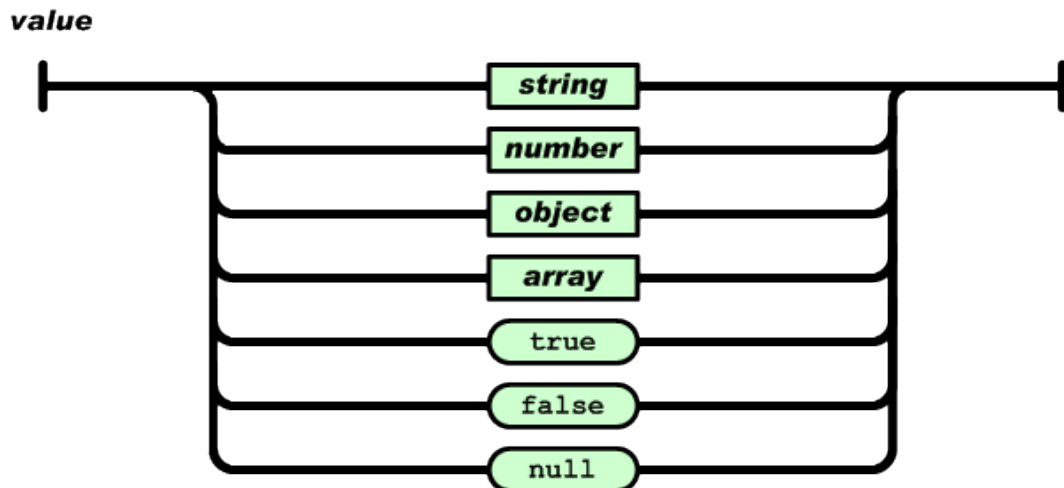
Dimana tag “{” menyatakan awal dari bentuk json itu sendiri sedangkan untuk selanjutnya dapat langsung diberikan identifier berupa string lalu “:” (titik dua), dan *value* dari *object* itu sendiri. Jika memungkinkan lebih dari 1 *object* maka dapat dihubungkan dengan tanda “,” (koma) dan jika sudah selesai harus diakhiri dengan tag “}”.



Gambar 2.3 JSON Array

(Sumber: <https://www.json.org/>, 2018)

Dimana tag “[” menyatakan awal dari bentuk json array, sedangkan untuk selanjutnya dapat langsung diberikan *value*, dimana *value* itu sendiri dapat berupa String, angka, *object*, array itu sendiri (nested array), Boolean, null seperti terlihat di Gambar 2.4 JSON Value



Gambar 2.4 JSON Value

(Sumber: <https://www.json.org/>, 2018)

2.6 Pengujian

Pengujian adalah sebuah cara yang digunakan untuk memastikan sistem yang telah dikembangkan sesuai dengan kebutuhan yang telah dideklarasikan dibagian awal dari pengembangan sebuah sistem. Dengan adanya pengujian diharapkan sistem yang dikembangkan tidak melenceng dari kebutuhan sistem tersebut. Pengujian juga memastikan bahwa setiap fungsi yang ada dalam sistem tersebut berjalan sesuai dengan kasus uji yang diberikan, dimana setiap kasus uji terkait dengan kebutuhan yang telah dideklarasikan diawal pengujian. Pengujian adalah serangkaian kegiatan yang dapat direncanakan sebelumnya dan dilakukan secara sistematis. Karena alasan ini, template untuk pengujian perangkat lunak — serangkaian langkah di mana penguji dapat menempatkan teknik desain uji spesifik dan metode pengujian — harus ditentukan untuk proses perangkat lunak. (Pressman, 2008)

2.6.1 Pengujian Fungsional

Pengujian Fungsional adalah bagian dari pengujian *Black Box testing* yang mana merupakan pengujian yang berfokus pada keluaran hasil dari masukan atau *input* yang diberikan oleh responden, secara sederhana black box testing merupakan proses menggunakan aplikasi untuk mengetahui apakah ada error atau ada fungsi yang tidak berjalan sesuai ekspektasi dari pengembangan. *Black box testing* merupakan teknik yang mengabaikan mekanisme internal dari sistem (kode / proses) dan hanya melihat apakah keluaran yang diberikan oleh sistem sesuai dengan keluaran yang diharapkan jika diberikan masukan tertentu. Pengujian black-box, juga disebut pengujian perilaku, berfokus pada persyaratan fungsional perangkat lunak. Yaitu, teknik pengujian black-box memungkinkan Anda untuk menentukan set kondisi input yang akan sepenuhnya menjalankan semua persyaratan fungsional untuk suatu program. Pengujian black-box bukan merupakan alternatif teknik white-box. Sebaliknya, itu adalah pendekatan

pelengkap yang kemungkinan akan mengungkap kelas kesalahan yang berbeda dari metode whitebox. Pengujian black-box mencoba menemukan kesalahan dalam kategori berikut: (1) fungsi yang salah atau yang hilang, (2) kesalahan antarmuka, (3) kesalahan dalam struktur data atau akses database eksternal, (4) kesalahan perilaku atau kinerja, dan (5) inisialisasi dan terminasi kesalahan. (Pressman, 2008)

2.6.2 Pengujian Kompatibilitas

Pengujian kompatibilitas adalah pengujian yang digunakan untuk memeriksa apakah perangkat lunak yang telah dikembangkan dapat berjalan pada perangkat keras, perangkat lunak, sistem operasi atau lingkungan yang berbeda dari lingkungan pengembangan. Pengujian kompatibilitas digunakan untuk menentukan aplikasi yang dikembangkan dapat berjalan lancar pada lingkungan lain sesuai yang diharapkan. Pengujian ini dapat dikatakan berhasil jika aplikasi yang dikembangkan dapat berjalan sesuai harapan pada lingkungan yang jelas berbeda dari lingkungan pengembang. Sebuah Aplikasi harus dapat digunakan di lingkungan yang berbeda (seperti berbeda perangkat keras, berbeda koneksi, berbeda perangkat lunak (sistem operasi), berbeda *browser*) dan harus didesain untuk kompatibel dengan semua (Pressman, 2008).

2.6.3 Pengujian Usabilitas

Pengujian usabilitas adalah pengujian yang berfokus pada tingkat kualitas dari aplikasi dan juga untuk menentukan seberapa mudah pengguna menggunakan dan memahami aplikasi yang sedang digunakan. Aplikasi dapat disebut berguna jika fungsionalitas dari aplikasi dapat dijalankan secara mudah, efektif, efisien, dan memuaskan dalam menjalankan dan mencapai tujuan dari aplikasi itu sendiri.

Pengujian ini menggunakan dua metode yaitu Success Rate berdasarkan skenario berdasarkan tujuan (*Goal / Task Based Scenario*) yang mana akan menentukan tingkat keberhasilan dari proses penggunaan aplikasi, jika tingkat keberhasilan dalam penggunaan aplikasi semakin baik setiap dilakukan pengujian maka dapat disimpulkan bahwa sistem dikatakan baik dalam penggunaannya, namun jika tingkat keberhasilannya semakin berkurang maka akan dikatakan bahwa sistem kurang berhasil dalam desain aplikasi. Pengalaman penggunaan internet rata-rata adalah sekali gagal, ketika pengguna mencoba untuk pertama kalinya, biasanya mereka gagal, sehingga dapat dikatakan kebanyakan *website* memiliki persentase keberhasilan dibawah 50% (Nielsen Norman Group, 2018).

BAB 3 METODOLOGI PENELITIAN

Pada bab metodologi penelitian ini akan dibahas metode yang digunakan dalam pengembangan perangkat lunak dan dijelaskan langkah-langkah dalam mengimplementasikan metode pada pengembangan perangkat lunak. Adapun diagram metodologi yang akan digunakan seperti dalam Gambar 3.1 sebagai berikut :



Gambar 3.1 Diagram Alir Metodologi

3.1 Studi Pustaka

Studi pustaka ini menjelaskan dasar-dasar teori yang bersumber dari buku, jurnal, halaman *web*, dokumentasi *framework*, *library*, serta bahasa pemrograman maupun sumber penelitian lain yang terkait yang digunakan

sebagai acuan untuk melakukan pengembangan aplikasi ini. Teori pendukung tersebut diantaranya :

1. Android SDK *Platform*
2. Firebase
3. Firebase Database
4. *JavaScript Object Notation* (JSON)
5. Pengujian
 1. Pengujian Fungsional
 2. Pengujian Kompabilitas
 3. Pengujian Usabilitas

3.2 Rekayasa Kebutuhan

Tahap ini akan menjelaskan terkait analisis dan kebutuhan dalam pengembangan system yang memanfaatkan firebase API pada perangkat bergerak. Tahap ini bertujuan untuk mendapatkan kebutuhan sistem yang nantinya akan dijadikan acuan untuk mengembangkan aplikasi *mobile tracer study* berbasis Android. Kebutuhan yang nanti didapat meliputi identifikasi actor (pengguna) aplikasi, kebutuhan fungsional maupun non fungsional.

Teknik elisitasi yang akan dilakukan adalah dengan diskusi atau wawancara secara langsung dengan narasumber yang berperan langsung dalam proses *tracer study* itu sendiri, seperti bagian kemahasiswaan dari fakultas, ketua jurusan, hingga dosen yang ikut ambil andil dalam proses *tracer study* itu sendiri. Sehingga nantinya didapatkan data dan kebutuhan yang dibutuhkan untuk tahap perancangan.

Hasil yang akan didapatkan dari rekayasa kebutuhan adalah kebutuhan fungsional dan non fungsional yang nantinya akan dilakukan pemodelan dengan menggunakan UML. Pemodelan system dengan menggunakan UML seperti *use case diagram* dan *use case scenario* yang berdasarkan pada kebutuhan fungsional yang ada dan telah didapatkan dari wawancara dengan narasumber. Penulis akan menuliskan juga aktor yang dibutuhkan dalam aplikasi *tracer study* untuk kemudian dijelaskan kebutuhan fungsional berdasarkan tiap aktor yang ada dalam aplikasi *tracer study*, dan kebutuhan non fungsional yang harus ada pada aplikasi *tracer study*.

Setelah kebutuhan didapatkan, dan telah direkayasa maka akan dibuat prototype dari kebutuhan yang didapatkan, yang nantinya prototype tersebut akan didiskusikan dengan pengguna atau klien yang nantinya akan berubah-ubah sesuai dengan keinginan pengguna atau klien, hasil akhir akan dituliskan sebagai hasil yang sudah disepakati antara pengembang dengan pengguna atau klien sehingga dalam proses pengembangan lebih lanjutnya tidak akan berulang ulang seperti proses pembuatan prototype.

3.2.1 Identifikasi Aktor

Dari daftar kebutuhan yang didapat pada proses elisitasi yang digunakan akan menghasilkan kebutuhan fungsional, kebutuhan non fungsional serta aktor aktor yang berperan langsung dalam berhubungan dengan sistem. Penjelasan identifikasi aktor akan berfungsi dalam identifikasi kebutuhan fungsional dan non fungsional nantinya sehingga lebih mudah memahami perbedaan yang ada pada kebutuhan fungsional dan kebutuhan non fungsional.

3.2.2 Identifikasi Kebutuhan

Dari daftar kebutuhan yang didapat pada proses elisitasi yang telah dilakukan sebelumnya, maka perlu sebuah penjelasan perbedaan antara kebutuhan fungsional, kebutuhan non fungsional, aktor yang berperan dalam pengembangan sistem, sehingga dibuat sebuah kode khusus yang menjelaskan perbedaan yang ada pada setiap kebutuhan, jika tidak ada identifikasi kebutuhan maka akan menyebabkan kebingungan pada pengembangan sistem itu sendiri, karena memungkinkan bentroknya kebutuhan sistem atau deklarasi kebutuhan berbeda yang mirip sehingga menjadi ambigu. Pada tahap ini akan dilakukan penjelasan tentang kode identifikasi yang digunakan dalam pengembangan sistem *tracer study*.

3.2.3 Kebutuhan Fungsional

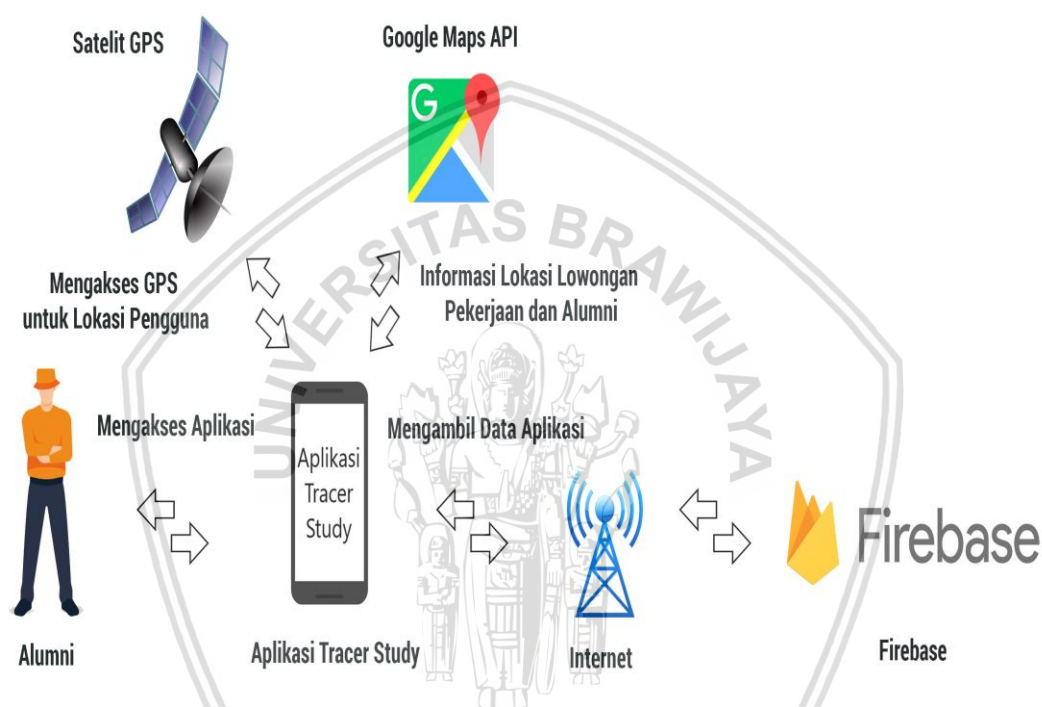
Kebutuhan fungsional adalah kebutuhan yang berhubungan langsung dengan fitur sistem yang harus ada dalam penyelesaian masalah, apa saja yang sistem berikan kepada pengguna aplikasi yang akan dirancang dan diimplementasikan dalam aplikasi. Kebutuhan fungsional pengguna adalah pernyataan dari pengguna tentang apa saja yang diinginkan ada dalam sistem itu sendiri. Dalam tahap ini, peneliti akan menentukan kebutuhan fungsional dari aplikasi *tracer study* yang akan dirancang dan diimplementasikan langsung dalam aplikasi *tracer study*. Kebutuhan fungsional pada aplikasi *tracer study* didapatkan dengan cara wawancara langsung dengan pihak kemahasiswaan, ketua jurusan, dosen yang bertanggung jawab pada proses *tracer study*, dan alumni.

3.2.4 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah kebutuhan yang tidak berhubungan langsung dengan pengguna, namun tetap menentukan kualitas layanan dari aplikasi itu sendiri. Kebutuhan non-fungsional dapat berupa batasan waktu akses aplikasi, batasan pengembangan proses, kualitas yang harus dipenuhi dalam pengembangan dan pengimplementasian sistem itu sendiri. Dalam metodologi penelitian tahap ini, peneliti akan menentukan kebutuhan non fungsional dari aplikasi *tracer study* yang akan dilakukan dengan wawancara langsung dengan pihak kemahasiswaan, ketua jurusan, dosen yang bertanggung jawab pada proses *tracer study*, dan alumni.

3.2.5 Pemodelan Kebutuhan

Tahap ini akan dilakukan setelah proses identifikasi kebutuhan telah selesai. Pada tahap ini, peneliti akan melakukan perancangan arsitektur aplikasi maupun antar muka. Perancangan arsitektur menggunakan teknik yang mengadopsi dari proses MVP (Model View Presenter) yang mana MVP adalah salah satu jenis pola perancangan aplikasi yang mana membedakan fungsi dari tampilan dengan fungsional yang berjalan dibelakang aplikasi. Dengan menggunakan MVP diharapkan dapat membuat sistem yang dapat lebih mudah membantu pembaca ataupun pengembang lainnya memahami alur hubungan antara tampilan, pengontrol (*controller*) dan entitas yang ada.



Gambar 3.2 Gambaran Umum Sistem

Dari gambar diatas menjelaskan pertukaran data system yang akan dibangun dengan backend dimana nantinya akan mengatur hak akses atau authentication pengguna *tracer study*. Data yang dikirimkan oleh system akan disimpan dalam *realtime database* yang dimiliki oleh cloud computing dari sistem yang disebut sebagai firebase. Firebase memiliki beberapa fungsi selain sebagai media penyimpanan data juga sebagai media *authentication*, *cloud storage*.

Dalam penelitian ini perancangan sistem yang akan digunakan adalah teknik pengembangan secara *object oriented*. Pengembangan *object oriented* biasanya membutuhkan bantuan diagram dalam pengembangannya namun pengembangan *object oriented* pada umumnya yang menggunakan *sequence diagram* dan *class diagram* dirasa kurang cocok untuk pengembangan aplikasi *mobile*, maka dengan membuat usecase diagram, rancangan *database*, dan perancangan *user experience* lebih cocok dalam pengembangan *object oriented*:

1. Use case Diagram

Use case diagram merupakan diagram yang menggambarkan hubungan interaksi antara aktor dengan fungsionalitas dari sistem. Diagram ini menjelaskan juga hubungan antara sebuah fungsionalitas dari sistem dengan fungsionalitas sistem lainnya dalam bentuk hubungan include dan extends.

2. Perancangan Data

Perancangan *data* digunakan untuk menentukan isi dan pengaturan tata letak data yang sangat dibutuhkan dalam proses rancangan sistem, perancangan *database* yang baik akan mendukung proses pengimplementasian nantinya.

3. Perancangan User Experience

Perancangan *User Experience* meliputi 3 bagian, yaitu desain antar muka, *screenflow*, serta desain tugas pengguna. Dalam perancangan desain antar muka nantinya akan digambarkan antar muka yang akan dikembangkan menjadi sebuah *user interface* nantinya. Sedangkan dalam perancangan *screenflow* ialah perancangan yang berhubungan tentang urutan dari halaman yang ditampilkan, dimana nantinya akan digambarkan tentang *screenflow* atau urutan tampilan yang berhubungan dengan fitur fitur aplikasi nantinya. Desain tugas pengguna adalah perancangan berupa tugas tugas apa saja yang akan dikerjakan oleh pengguna aplikasi yaitu alumni.

3.2.6 Analisis Pertanyaan Tracer Study

Tahap ini akan dilakukan saat proses pemodelan kebutuhan selesai, yang mana nantinya akan dilakukan analisis terhadap pertanyaan yang perlu dimasukkan kedalam daftar pertanyaan sistem *tracer study*, yang mana pertanyaan yang dimaksud adalah pertanyaan yang sudah didapatkan saat analisis kebutuhan yaitu saat elisitasi kebutuhan.

3.3 Perancangan

Tahap perancangan adalah tahap yang mengubah rekayasa kebutuhan menjadi sebuah rancangan pengembangan perangkat lunak, yang mana pada tahap ini akan dibuat *activity diagram*, perancangan kode, perancangan data, perancangan antarmuka yang terdiri dari *screenflow diagram* dan perancangan antarmuka itu sendiri, serta perancangan tugas pengguna yang akan berisi apa saja yang dapat dilakukan oleh seorang pengguna aplikasi.

3.3.1 Diagram Activity

Pada tahap ini, akan dilakukan sebuah perancangan diagram yang akan menghubungkan hubungan sistem dan pengguna sistem secara dinamis yang artinya akan terpengaruh berdasarkan waktu dalam proses penggunaan aplikasi atau sistem itu sendiri.

3.3.2 Perancangan Data

Pada tahap ini, akan dilakukan perancangan berbentuk skema json, yang mana akan menghubungkan *object-object* yang saling terhubung pada proses implementasi data nantinya. Skema json yang terdapat pada bagian ini adalah bentuk sederhana dari hasil yang nantinya diharapkan implementasi sesuai dengan skema json yang dijelaskan pada tahap ini.

3.3.3 Perancangan Antarmuka

Pada tahap ini, akan dilakukan sebuah perancangan tampilan yang mana akan berupa screenflow diagram, atau hubungan antar tampilan, dan akan dilakukan perancangan antarmuka itu sendiri, atau yang biasa disebut sebagai mock up atau tampilan kasar dari antarmuka. Lalu akan dilanjutkan dengan proses perancangan tugas pengguna. Pada tahap ini, juga akan dilakukan perancangan yang melanjutkan hasil dari analisis *user experience* yang telah dilakukan, dimana analisis *user experience* akan menjabarkan bagaimana bentuk tampilan yang baik dan memudahkan pengguna sesuai dengan pedoman yang telah ada sebelumnya.

3.3.4 Screenflow Diagram

Pada tahap ini, akan dilakukan perancangan yang saling mengaitkan hubungan antara tampilan yang satu dengan yang lain, dengan hubungan sebuah garis, yang mana jika suatu tombol dapat ditekan maka tampilan apa yang akan dipanggil selanjutnya, atau tampilan apa yang akan ditampilkan setelah sebuah tombol ditekan.

3.3.5 Perancangan Tugas Pengguna

Pada tahap ini, akan dilakukan perancangan skenario yang akan digunakan sebagai dasar dalam penggunaan, dan menjadi tahap yang melanjutkan dari analisis kebutuhan non fungsional, dimana pada tahap ini akan diberikan tampilan yang akan diberikan rancangan apa yang akan dilakukan oleh sistem jika sebuah tombol atau tampilan ditekan. Pada tahap ini juga akan menjelaskan tentang usability yang memungkinkan dari tabel kebutuhan non fungsional.

3.4 Implementasi

Tahap ini akan dilakukan setelah proses perancangan perangkat lunak telah selesai, dan tahap ini pula akan menjabarkan detail implementasi yang telah diimplementasikan sesuai dengan pedoman pengembangan aplikasi Android yang telah dipelajari dalam mata kuliah pengembangan aplikasi perangkat bergerak dengan menggunakan bahasa Java, dan IDE Android Studio.

Implementasi dalam sisi *server* akan menggunakan api yang disediakan oleh firebase seperti *realtime database*, *storage*, serta *authentication*. Sedangkan dalam pengembangan sisi aplikasi *mobile* akan menggunakan sensor lokasi (gps), sensor kamera, serta memanfaatkan Android Studio dalam pengembangan *user interface* dan dalam implementasi aplikasi.

3.5 Pengujian

Setelah tahap implementasi selesai dilakukan, maka akan dilakukan pengujian yang bertujuan untuk mengetahui apakah aplikasi yang telah dibangun telah sesuai dengan kebutuhan perangkat lunak yang telah didapatkan pada fase analisis kebutuhan, pengujian yang digunakan adalah pengujian fungsional, kompatibilitas dan pengujian *usabilitas*.

1. Pengujian Fungsional

Pengujian ini meliputi mengujian fitur fitur aplikasi secara umum menggunakan metode black box. Teknik yang digunakan dalam metode black box ini adalah teknik *equivalence partitioning* yang mana akan memetakan mana saja yang merupakan masukan dari pengguna yang memungkinkan untuk dijadikan masukan. Masukan yang mungkin dijadikan masukan pada sistem hanya berupa nilai nilai yang menjadi batasan saja, jika yg dimungkinkan adalah 1 sampai 10, maka masukan yang dimasukkan adalah 0, 1,2 dan 9, 10, 11 yang mana seharusnya 0 dan 11 seharusnya gagal, dan 1 hingga 10 seharusnya berhasil.

2. Pengujian Kompatibilitas

Pengujian kompatibilitas adalah pengujian yang meliputi pengujian pada lingkungan perangkat keras perangkat lunak, sistem operasi, atau jaringan yang berbeda dan diharapkan hasil dari pengujian ini adalah lingkungan perangkat keras, perangkat lunak, sistem operasi atau jaringan yang berbeda dapat digunakan untuk menjalankan aplikasi. Aplikasi *tracer study* akan diuji dengan dijalankan di perangkat berbeda, dengan jaringan berbeda, dan sistem operasi berbeda. Jika aplikasi tidak mengalami masalah ketika dilakukan pengujian maka aplikasi dikatakan lulus uji.

3. Pengujian Usabilitas

Pengujian Usabilitas adalah pengujian yang berfokus pada tingkat kualitas dari aplikasi dan juga untuk menentukan seberapa mudah pengguna menggunakan dan memahami aplikasi yang sedang digunakan. Aplikasi dapat disebut usable (berguna) jika fungsionalitas dari aplikasi dapat dijalankan secara mudah, efektif, efisien, dan memuaskan dalam menjalankan dan mencapai tujuan dari aplikasi itu sendiri. Pengujian Usabilitas menggunakan teknik pengujian pengisian form pada pengguna yang baru menggunakan aplikasi ini, bagaimana *rating* yang diberikan oleh pengguna dengan menjawab jawaban yang menggunakan teknik pengujian usabilitas dengan teknik form *System Usability Scale* (SUS), dalam teknik pengujian ini, pernyataan ganjil akan dilakukan perhitungan yaitu hasil yang diberikan oleh penguji dikurangi satu dari hasil *response* yang diberikan oleh pengguna yang melakukan uji usabilitas, sedangkan untuk pernyataan genap akan dilakukan perhitungan lima dikurangi *response* yang diberikan oleh pengguna yang melakukan uji usabilitas, skala yang diberikan adalah skala nol sampai dengan empat, jika semua pertanyaan sudah dijawab oleh pengguna yang melakukan uji usabilitas, maka jumlahnya dikalikan

dengan dua koma lima (2,5) untuk mengkonversi nilai menjadi skala 0 – 100. Jumlah pertanyaan dalam pertanyaan SUS ada 10 pertanyaan dimana pertanyaan ganjil adalah pertanyaan yang *positive*, sedangkan pertanyaan genap adalah pertanyaan yang berunsur *negative*. Dalam prosesnya, jika rata-rata hasil diatas 68, maka akan dianggap lulus uji, dan dianggap berhasil, sedangkan jika hasil rata rata dibawah 68, maka akan dikatakan gagal atau dibawah rata-rata. (Sauro, 2011)

3.6 Analisis Hasil Pengujian

Tahap ini berisikan analisis hasil pengujian setelah pengujian selesai dilakukan pada aplikasi *mobile tracer study* dengan menggunakan *platform* Android dengan minimal API 16. Lalu dilanjutkan dengan penarikan kesimpulan dan saran.

Analisis hasil pengujian akan menghasilkan jawaban dari rumusan masalah yang telah dikemukakan oleh penulis, sehingga penulis harus menganalisis:

1. Bagaimanakah hasil hari pengujian fungsional yang telah dilakukan?
2. Bagaimanakah hasil hari pengujian kompatibilitas yang telah dilakukan?
3. Bagaimanakah hasil hari pengujian usabilitas yang telah dilakukan?

3.7 Penarikan Kesimpulan dan Saran

Tahap ini berisikan kesimpulan akhir setelah pengerjaan aplikasi *tracer study* menggunakan *platform* Android yang menjawab rumusan masalah yang telah dijabarkan sebelumnya, lalu dilanjutkan dengan pemberian saran untuk pengembangan selanjutnya supaya dapat terus dikembangkan diwaktu yang akan datang.

BAB 4 REKAYASA KEBUTUHAN

4.1 Gambaran Umum Sistem

Sistem *tracer study* adalah sistem yang menghubungkan antara mahasiswa yang sudah lulus atau sudah mendapatkan julukan alumni dari suatu instansi pendidikan dengan alumni yang lain sehingga memiliki hubungan antara satu alumni dan alumni yang lain. Alumni dapat saling memberikan informasi tentang lowongan pekerjaan kepada alumni yang lain. Alumni juga dapat memberikan informasi pekerjaan alumni saat ini, serta alumni dapat melihat informasi pekerjaan alumni juga, dan alumni juga dapat memberikan informasi pendidikan serta melihat informasi pendidikan yang telah diisikan oleh alumni itu sendiri.





Gambar 4.1 Story Board Tracer Study

Dalam gambar 4.1 menjelaskan tentang bagaimana nantinya mahasiswa yang lulus atau dapat dikatakan alumni dapat berinteraksi dengan sistem, kondisi seperti apa sistem dapat berguna bagi mereka. Berikut ini penjelasan gambar 4.1 per scene yang telah digambarkan, gambar dimulai dari kiri atas lalu kekanan dan diakhiri pada bagian kanan bawah:

1. Mahasiswa lulus sehingga mendapatkan julukan sebagai alumni dari suatu instansi pendidikan.
2. Alumni diharapkan mengakses sistem *tracer study* yang telah disediakan oleh instansi pendidikan.
3. Alumni mengisi data yang dibutuhkan oleh sistem *tracer study*.
4. Alumni dapat mengirim pesan kepada alumni yang lain, sehingga lebih mudah dalam berhubungan.
5. Alumni dapat mencari lowongan pekerjaan yang tersedia pada sistem *tracer study* yang akan dibangun.
6. Alumni yang telah mendapatkan pekerjaan, dapat mengisi data pekerjaan alumni, namun untuk alumni yang melanjutkan studi dapat mengisi data pendidikan alumni, jika telah lulus, alumni dapat mengisi informasi data pekerjaan alumni.

4.2 Analisis Kebutuhan

Tahap analisis kebutuhan pada penelitian ini dilakukan dengan metode wawancara. Wawancara dilakukan dengan pihak kemahasiswaan Fakultas Ilmu Komputer, Universitas Brawijaya, dan tim *tracer study* dari pihak dosen di Fakultas Ilmu Komputer, Universitas Brawijaya. Wawancara dilakukan untuk mendapatkan data yang berhubungan dengan *tracer study* yang pernah

dilakukan pihak kemahasiswaan Fakultas Ilmu Komputer, Universitas Brawijaya, dalam hal ini didapatkan jenis-jenis pengguna seperti alumni dan pengguna dari pihak atasan atau pihak perusahaan yang mempekerjakan alumni. Wawancara yang dilakukan dengan pihak dosen, didapatkan jenis-jenis data yang dibutuhkan dalam untuk proses penilaian instansi pendidikan seperti waktu tunggu alumni untuk waktu kerja, dan kesesuaian pekerjaan yang didapatkan oleh alumni dengan pendidikan yang ditempuh di Fakultas Ilmu Komputer, Universitas Brawijaya. Hasil yang didapatkan pada saat wawancara, didapatkan beberapa kebutuhan primer atau kebutuhan utama untuk dikembangkan pada aplikasi *tracer study* yaitu pertanyaan yang harus ditanyakan pada pertanyaan pekerjaan dan pertanyaan data pendidikan alumni, serta pertanyaan lowongan pekerjaan alumni.

4.3 Identifikasi Aktor

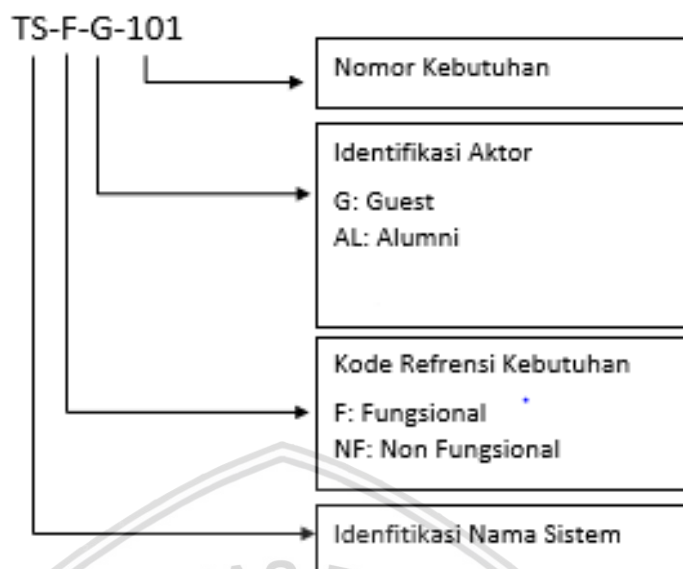
Pada tahapan ini, dilakukan identifikasi aktor yang ada dalam proses pengembangan aplikasi *tracer study*, sehingga dapat memberikan gambaran tentang siapa saja yang terlibat dalam sistem. Hasil dari identifikasi aktor dapat dilihat dalam Tabel 4.1:

Tabel 4.1 Daftar Aktor

| Nama Aktor | Deskripsi Aktor |
|--------------|---|
| <i>Guest</i> | Aktor umum yang menggunakan sistem namun memiliki hak akses yang sangat terbatas karena belum memiliki banyak akses terhadap sistem |
| Alumni | Aktor yang dapat menggunakan sistem dan memiliki hak akses lebih banyak daripada <i>guest</i> karena telah memiliki akses terhadap sistem |

4.4 Identifikasi Kebutuhan

Identifikasi kebutuhan adalah sebuah aturan yang digunakan dalam rekayasa kebutuhan yang dilakukan. Identifikasi kebutuhan yang ada pada sistem *tracer study* menggunakan aturan penomoran yang diambil dari nama sistem seperti sistem penomoran pada umumnya, yang menggunakan bagian awal (inisial) nama sistem itu sendiri, seperti *tracer study* (TS). Setelah singkatan nama aplikasi dilanjutkan dengan tingkat kebutuhan yang berhubungan dengan sistem, seperti penggunaan singkatan F untuk fungsional dan penggunaan singkatan NF untuk non fungsional, lalu diikuti dengan kode dari aktor itu sendiri, jika *guest* maka akan menggunakan kode G, atau jika alumni akan menggunakan AL. Lalu terakhir akan diikuti oleh nomor kebutuhan yang berupa nilai kelipatan dari 100, dan akan dijelaskan oleh nilai anak dengan nilai satuan dari nilai yang ada, setiap bagian akan di pisahkan dengan tanda – (strip / minus). Untuk lebih jelasnya terdapat dalam Gambar 4.2 berikut:



Gambar 4.2 Aturan Penomoran Kebutuhan

4.4.1 Kebutuhan Fungsional

Kebutuhan fungsional sistem yang didapatkan dari hasil analisis kebutuhan dalam bentuk wawancara dapat dilihat pada tabel 4.2 Berikut:

Tabel 4.2 Daftar Kebutuhan Fungsional *Guest*

| No | Kode | Nama Fungsional | Deskripsi |
|----|------------|-----------------|--|
| 1 | TS-F-G-101 | <i>Login</i> | Sistem harus dapat memberikan layanan bagi <i>guest</i> untuk melakukan otentikasi diri dengan memasukkan <i>email</i> , <i>nim</i> , serta <i>password</i> |
| 2 | TS-F-G-201 | <i>Register</i> | Sistem harus mampu memberikan layanan bagi <i>guest</i> untuk melakukan <i>register</i> atau mendaftar pada sistem untuk mendapatkan akses terhadap fungsionalitas alumni yang ada pada sistem |

Kebutuhan fungsional yang dimiliki oleh alumni berbeda dengan kebutuhan yang dimiliki oleh *guest*, tabel 4.3 menjelaskan tentang daftar dari kebutuhan fungsional yang dimiliki oleh alumni:

Tabel 4.3 Daftar Kebutuhan Fungsional Alumni

| No | Kode | Nama Fungsi | Deskripsi |
|----|-------------|-------------------------------|---|
| 1 | TS-F-AL-101 | Kirim Pesan | Sistem harus mampu memberikan layanan bagi alumni untuk mengirimkan pesan ke alumni yang lain |
| 2 | TS-F-AL-201 | Melihat Informasi Pekerjaan | Sistem harus mampu memberikan layanan bagi alumni untuk melihat informasi lowongan pekerjaan yang sedang dibuka sekarang |
| 3 | TS-F-AL-301 | Ubah Informasi Diri Alumni | Sistem harus mampu memberikan layanan bagi alumni untuk mengubah data yang telah dimasukkan dalam halaman registrasi dengan data yang telah dimasukkan sudah terjawab sesuai dengan jawaban alumni sebelumnya |
| 4 | TS-F-AL-401 | Masukkan Informasi Pekerjaan | Sistem harus mampu memberikan layanan bagi alumni untuk memasukkan informasi pekerjaan alumni |
| 5 | TS-F-AL-501 | Masukkan Informasi Pendidikan | Sistem harus mampu memberikan layanan bagi alumni untuk memasukkan informasi pendidikan alumni |
| 6 | TS-F-AL-601 | Masukkan Lowongan Kerja | Sistem harus mampu memberikan layanan bagi alumni untuk memasukkan informasi lowongan pekerjaan yang sedang dibuka di perusahaan alumni. |
| 7 | TS-F-AL-701 | Lihat Data Pendidikan | Sistem harus mampu memberikan layanan bagi alumni untuk melihat informasi data pendidikan alumni |
| 8 | TS-F-AL-801 | Lihat Data Pekerjaan | Sistem harus mampu memberikan layanan bagi alumni untuk melihat informasi data pekerjaan alumni |

4.4.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional adalah kebutuhan yang tidak berhubungan langsung dengan pengguna sistem (baik *guest* maupun alumni) namun menentukan kualitas dan kepuasan pengguna terhadap penggunaan sistem. Kebutuhan ini tidak dibutuhkan secara langsung oleh pengguna, namun mempengaruhi pandangan pengguna terhadap sistem. Kebutuhan non fungsional dijelaskan pada tabel 4.4:

Tabel 4.4 Kebutuhan Non Fungsional

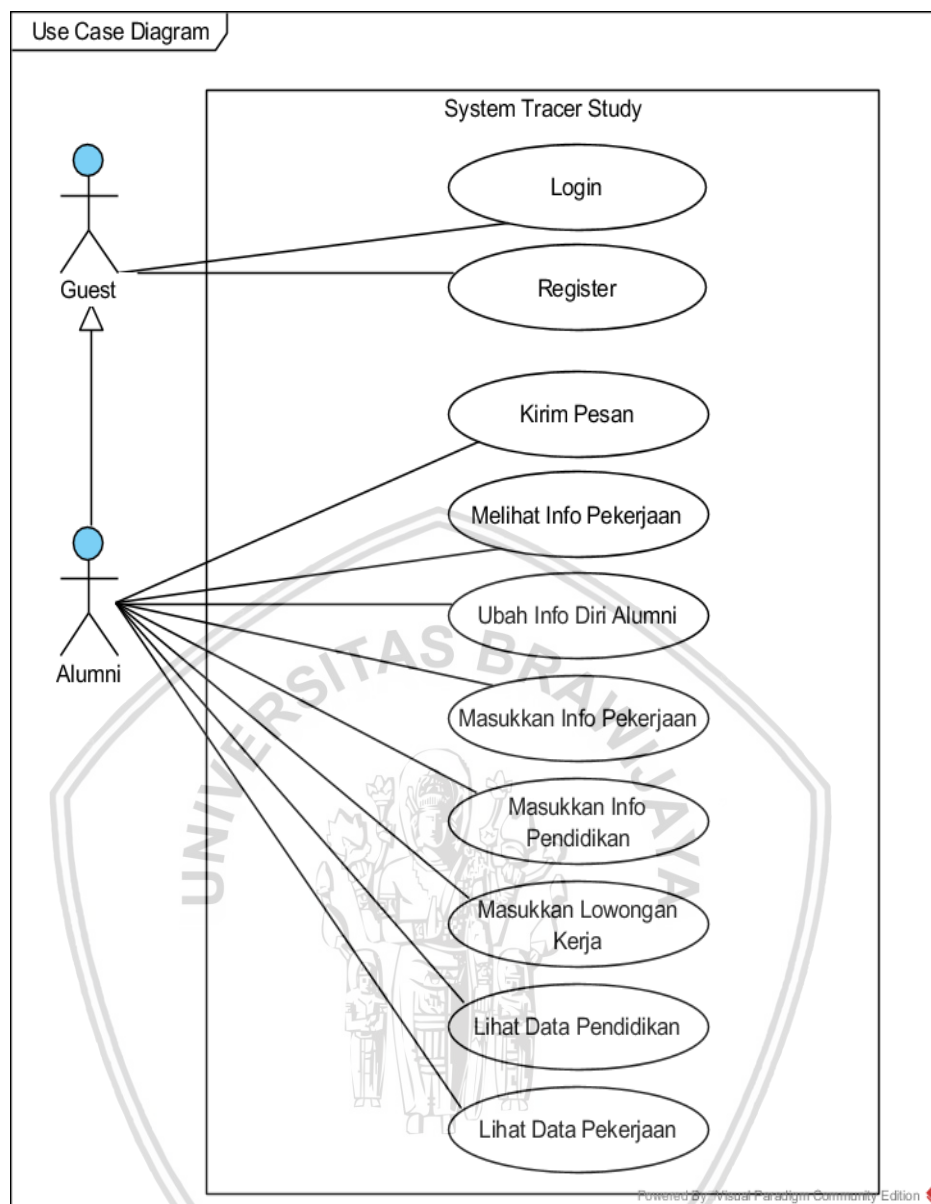
| Parameter | Deskripsi Kebutuhan |
|----------------|--|
| Usabilitas | Rancangan antarmuka mudah digunakan oleh alumni dengan menggunakan kaidah usabilitas yaitu <i>learnability</i> , efisiensi, <i>memorability</i> , <i>error</i> , <i>satisfaction</i> dengan target tingkat usabilitas minimal 68% (Sauro, 2011). |
| Kompatibilitas | Rancangan aplikasi harus bisa digunakan dalam perangkat keras yang memiliki sistem operasi Android minimal API 16 atau Jelly Bean. |

4.5 Pemodelan Kebutuhan

Pemodelan kebutuhan adalah proses memodelkan kebutuhan yang telah diidentifikasi dan dianalisis dari kebutuhan fungsional yang telah dijelaskan sebelumnya sehingga didapatkan diagram *use case* dan skenario *use case*.

4.5.1 Diagram Use Case

Diagram use case adalah diagram yang merepresentasikan apa saja yang dapat dilakukan oleh aktor terhadap sistem yang dilihat dari tampak luar sistem. Diagram use case didapatkan dari kebutuhan fungsional yang ada didalam tabel 4.2 dan 4.3, kebutuhan maka didapatkan use case yang dapat dilihat dalam gambar 4.3:



Gambar 4.3 Use Case Diagram

4.5.2 Use Case Scenario

Setiap *use case* yang ada di gambar 4.3 akan dijelaskan lebih detail dan lebih lengkap pada tabel-tabel berikut yang akan lebih menjelaskan tentang *use case* tersebut. Pada tabel 4.5 hingga tabel 4.14, akan menjelaskan secara detail tentang skenario yang akan *guest* lakukan saat mengakses sistem *tracer study*.

Tabel 4.5 Skenario *Login* (TS-F-G-101)

| | |
|----------------------|---|
| Item | TS-F-G-101 |
| Nama <i>Use Case</i> | <i>Login</i> |
| Deskripsi | Sistem harus dapat memberikan layanan bagi <i>guest</i> untuk melakukan otentikasi diri dengan memasukkan <i>email</i> , <i>nim</i> serta <i>password</i> |
| Aktor | <i>Guest</i> |
| Pra-Kondisi | <i>Guest</i> berada di halaman <i>login</i> |
| Tindakan | <ol style="list-style-type: none"> 1. <i>Guest</i> memasukkan data berupa <i>email</i> serta <i>password</i> di halaman <i>login</i> 2. <i>Guest</i> menekan tombol <i>login</i> yang ada di bagian bawah <i>form</i> 3. Sistem melakukan otentikasi pengguna dengan menggunakan API yang disediakan oleh Firebase |
| Alternatif | <ol style="list-style-type: none"> 1. Sistem memunculkan informasi jika <i>login</i> gagal karena <i>email</i> tidak diisi 2. Sistem memunculkan informasi jika <i>login</i> gagal karena <i>password</i> tidak diisi 3. Sistem memunculkan informasi jika <i>login</i> gagal karena <i>email</i> atau <i>password</i> tidak terdaftar |
| Post-Kondisi | Otentikasi berhasil, pengguna diarahkan ke halaman utama |

Tabel 4.6 Skenario *Register* (TS-F-G-201)

| | |
|----------------------|---|
| Item | TS-F-G-201 |
| Nama <i>Use Case</i> | <i>Register</i> |
| Deskripsi | Sistem harus mampu memberikan layanan bagi <i>guest</i> untuk melakukan <i>register</i> atau mendaftar pada sistem untuk mendapatkan akses terhadap fungsionalitas alumni yang ada pada sistem |
| Aktor | <i>Guest</i> |
| Pra-Kondisi | <i>Guest</i> berada di halaman <i>register</i> |
| Tindakan | <ol style="list-style-type: none"> 1. <i>Guest</i> memasukkan data berupa <i>email</i>, <i>nim</i>, <i>password</i>, nama lengkap, nama panggilan, alamat, tanggal kelulusan, jenis kelamin, no telpon, program studi, tahun masuk dan tahun lulus di halaman <i>register</i> 2. <i>Guest</i> menekan tombol <i>register</i> yang ada di bagian bawah |

| | |
|--------------|--|
| | <p><i>form</i></p> <ol style="list-style-type: none"> 3. Sistem melakukan otentikasi <i>guest</i> dengan menggunakan API yang disediakan oleh firebase 4. Jika otentikasi berhasil, maka data alumni akan disimpan dalam <i>database</i> sistem dan firebase <i>database</i> |
| Alternatif | <ol style="list-style-type: none"> 1. Sistem memunculkan informasi <i>email</i> tidak valid jika <i>email</i> yang dimasukkan tidak valid 2. Sistem memunculkan informasi nim tidak valid jika nim yang dimasukkan tidak sesuai format panjang nim yang ditentukan 3. Sistem memunculkan informasi jika <i>register</i> gagal |
| Post-Kondisi | Otentikasi dan penyimpanan data alumni berhasil, alumni diarahkan ke halaman utama |

Tabel 4.7 Skenario Kirim Pesan (TS-F-AL-101)

| | |
|----------------------|--|
| Item | TS-F-AL-101 |
| Nama <i>Use Case</i> | Kirim Pesan |
| Deskripsi | Sistem harus mampu memberikan layanan bagi alumni untuk mengirimkan pesan ke alumni lain |
| Aktor | Alumni |
| Pra-Kondisi | Alumni telah berada di halaman utama |
| Tindakan | <ol style="list-style-type: none"> 1. Alumni menekan tombol tambah yang ada diatas halaman utama 2. Alumni memasukkan identitas penerima pesan berupa nama dan nim penerima 3. Alumni memasukkan pesan yang akan dikirim 4. Alumni menekan tombol kirim yang ada disebelah tempat pengetikan pesan |
| Alternatif | <ol style="list-style-type: none"> 1. Sistem memunculkan informasi gagal membuat pesan jika data yang dimasukkan tidak sesuai dengan validasi nim |
| Post-Kondisi | Pesan terkirim |

Tabel 4.8 Skenario Melihat Lowongan Pekerjaan (TS-F-AL-201)

| | |
|----------------------|----------------------------|
| Item | TS-F-AL-201 |
| Nama <i>Use Case</i> | Melihat Lowongan Pekerjaan |

| | |
|--------------|--|
| Deskripsi | Sistem harus mampu memberikan layanan bagi alumni untuk melihat informasi lowongan pekerjaan yang sedang dibuka sekarang |
| Aktor | Alumni |
| Pra-Kondisi | Alumni telah berada di halaman lowongan pekerjaan |
| Tindakan | 1. Alumni dapat menekan salah satu lowongan pekerjaan tersedia untuk melihat rincian dari lowongan pekerjaan tersebut |
| Alternatif | 1. Sistem memunculkan pesan informasi pekerjaan kosong jika tidak ada informasi lowongan pekerjaan |
| Post-Kondisi | Alumni melihat rincian informasi pekerjaan |

Tabel 4.9 Skenario Ubah Informasi Diri Alumni (TS-F-AL-301)

| | |
|----------------------|---|
| Item | TS-F-AL-301 |
| Nama <i>Use Case</i> | Ubah Informasi Diri Alumni |
| Deskripsi | Sistem harus mampu memberikan layanan bagi alumni untuk mengubah data yang telah dimasukkan dalam halaman registrasi dengan data yang telah dimasukkan sudah terjawab sesuai dengan jawaban alumni sebelumnya |
| Aktor | Alumni |
| Pra-Kondisi | Alumni telah berada di halaman ubah data diri |
| Tindakan | 1. Alumni mengubah data yang diinginkan dari semua data yang telah ada 2. Alumni menekan tombol simpan untuk menyimpan hasil perubahan yang telah dilakukannya |
| Alternatif | 1. Sistem memunculkan pesan jawaban harus diisi jika ada jawaban yang tidak diisi oleh alumni |
| Post-Kondisi | Sistem berhasil menyimpan perubahan yang dilakukan oleh alumni, alumni dialihkan ke halaman utama |

Tabel 4.10 Skenario Memasukkan Informasi Pekerjaan (TS-F-AL-401)

| | |
|----------------------|---|
| Item | TS-F-AL-401 |
| Nama <i>Use Case</i> | Masukkan Informasi Pekerjaan |
| Deskripsi | Sistem harus mampu memberikan layanan bagi alumni untuk memasukkan informasi pekerjaan alumni |

| | |
|--------------|---|
| Aktor | Alumni |
| Pra-Kondisi | Alumni telah berada di halaman masukkan informasi pekerjaan |
| Tindakan | 1. Alumni mengisi informasi yang ditanyakan pada halaman pertanyaan 2. Alumni menekan tombol simpan |
| Alternatif | 1. Sistem memunculkan pesan jawaban harus diisi jika ada jawaban yang tidak diisi oleh alumni |
| Post-Kondisi | Sistem berhasil menyimpan jawaban dari pertanyaan pekerjaan yang dilakukan oleh alumni, alumni dialihkan ke halaman utama |

Tabel 4.11 Skenario Memasukkan Informasi Pendidikan (TS-F-AL-501)

| | |
|----------------------|--|
| Item | TS-F-AL-501 |
| Nama <i>Use Case</i> | Masukkan Informasi Pendidikan |
| Deskripsi | Sistem harus mampu memberikan layanan bagi alumni untuk memasukkan informasi pendidikan alumni |
| Aktor | Alumni |
| Pra-Kondisi | Alumni telah berada di halaman masukkan informasi pendidikan |
| Tindakan | 1. Alumni mengisi informasi yang ditanyakan pada halaman pertanyaan 2. Alumni menekan tombol simpan |
| Alternatif | 1. Sistem memunculkan pesan jawaban harus diisi jika ada jawaban yang tidak diisi oleh alumni |
| Post-Kondisi | Sistem berhasil menyimpan jawaban dari pertanyaan pendidikan yang dilakukan oleh alumni, alumni dialihkan ke halaman utama |

Tabel 4.12 Skenario Memasukkan Lowongan Kerja (TS-F-AL-601)

| | |
|----------------------|--|
| Item | TS-F-AL-601 |
| Nama <i>Use Case</i> | Masukkan Lowongan Kerja |
| Deskripsi | Sistem harus mampu memberikan layanan bagi alumni untuk memasukkan informasi lowongan pekerjaan yang sedang dibuka di perusahaan alumni. |

| | |
|--------------|---|
| Aktor | Alumni |
| Pra-Kondisi | Alumni telah berada di halaman masukkan lowongan kerja |
| Tindakan | 1. Alumni mengisi informasi yang ditanyakan pada halaman pertanyaan 2. Alumni menekan tombol simpan |
| Alternatif | 1. Sistem memunculkan pesan jawaban harus diisi jika ada jawaban yang tidak diisi oleh alumni |
| Post-Kondisi | Sistem berhasil menyimpan jawaban dari pertanyaan lowongan pekerjaan yang dilakukan oleh alumni, alumni dialihkan ke halaman lowongan pekerjaan |

Tabel 4.13 Skenario Melihat Data Pendidikan (TS-F-AL-701)

| | |
|----------------------|--|
| Item | TS-F-AL-701 |
| Nama <i>Use Case</i> | Lihat Data Pendidikan |
| Deskripsi | Sistem harus mampu memberikan layanan bagi alumni untuk melihat informasi data pendidikan alumni |
| Aktor | Alumni |
| Pra-Kondisi | Alumni telah berada di halaman profil |
| Tindakan | 1. Alumni menekan ikon lihat data pendidikan |
| Alternatif | 1. Sistem memunculkan pesan data tidak ditemukan jika data pendidikan tidak ditemukan |
| Post-Kondisi | Sistem menampilkan data pendidikan alumni |

Tabel 4.14 Skenario Melihat Data Pekerjaan (TS-F-AL-801)

| | |
|----------------------|---|
| Item | TS-F-AL-801 |
| Nama <i>Use Case</i> | Lihat Data Pekerjaan |
| Deskripsi | Sistem harus mampu memberikan layanan bagi alumni untuk melihat informasi data pekerjaan alumni |
| Aktor | Alumni |
| Pra-Kondisi | Alumni telah berada di halaman profil |
| Tindakan | 1. Alumni menekan ikon lihat data pekerjaan |
| Alternatif | 1. Sistem memunculkan pesan data tidak ditemukan jika data pendidikan tidak ditemukan |
| Post-Kondisi | Sistem menampilkan data pekerjaan alumni |

4.6 Analisis Data

Analisis data membahas tentang data-data yang dibutuhkan dalam proses perancangan data yang nantinya akan dibuat sebagai implementasi data. Bentuk analisis data berikut, berisi data yang dibutuhkan dalam pengembangan sistem *tracer study*, yang mana nantinya tidak akan terdapat tambahan data ketika dilakukan perancangan dan implementasi sistem *tracer study*. Analisis data yang akan digunakan pada sistem *tracer study* adalah sebagai berikut:

1. Data pengguna *tracer study* alumni akan berupa data data yang berhubungan dengan alumni, seperti *email*, *nim*, *password*, alamat alumni, jenis kelamin, nama lengkap, nama panggilan, program studi, bulan lulus, tahun lulus, serta tahun masuk alumni.
2. Data pekerjaan terdiri dari nomor induk mahasiswa yang mengisi, bulan kerja, tahun kerja, informasi status pekerjaan, informasi bidang pekerjaan, kesesuaian dengan keahlian pada saat lulus, nama perusahaan, jabatan, serta pendapatan perbulan yang didapat.
3. Data pendidikan terdiri dari nomor induk alumni, universitas tempat alumni melanjutkan pendidikan, kota universitas tempat alumni melanjutkan pendidikan, negara tempat alumni melanjutkan pendidikan, tahun lulus alumni, program studi, gelar yang diperoleh oleh alumni dari studinya, serta jenjang pendidikan alumni yaitu S1, S2 atau S3.

Sistem menggunakan database dari Firebase karena kebutuhan kirim pesan, dimana membutuhkan fitur yang sediakan oleh Firebase, yaitu Firebase Cloud Messaging (FCM). Dimana dengan fitur tersebut sistem dapat langsung memberikan data dan notifikasi ke dalam perangkat bergerak.

4.7 Analisis User Experience

Analisis *user experience* meliputi 3 bagian, yaitu analisis antar muka, *screenflow*, serta analisis tugas pengguna:

1. Analisis antarmuka

Pada desain antarmuka akan digambarkan tampilan yang nantinya akan diimplementasikan, desain antarmuka yang dibuat adalah semua bagian inti dari aplikasi, serta halaman halaman yang bisa diakses oleh alumni sebagai pengguna aplikasi *tracer study*. Desain antarmuka ini nantinya diharapkan akan sama dengan implementasi, namun tidak menutup kemungkinan hasil implementasi yang ada akan sedikit berbeda dengan hasil perancangan yang sudah dilakukan karena desain antarmuka yang dibangun tidak semudah dalam implementasi yang dilakukan, jika terdapat perubahan pada tampilan desain antar muka, nantinya akan dilakukan juga perubahan pada desain *screenflow* yang telah atau belum dibuat.

2. Analisis Screenflow

Halaman yang telah dijelaskan pada desain antar muka, nantinya akan digambarkan menggunakan garis hubung yang membantu memahami jalannya aplikasi nantinya ketika diimplementasikan, agar mendapat pandangan tentang alur sistem *tracer study* saat implemetasi, jika terdapat tambahan atau perubahan pada desain antarmuka maka akan dilakukan proses perubahan agar menyesuaikan dengan desain antarmuka yang dibangun. Setelah dihubungkan dengan menggunakan garis hubung, diharapkan nantinya implementasi antarmuka dan jalannya aplikasi dapat mengikuti desain yang telah dibuat.

3. Analisis Tugas Pengguna

Desain tugas pengguna adalah jenis jenis tugas yang akan dibebankan kepada alumni sebagai pengguna, dimana alumni dapat memiliki beberapa tugas utama yang harus didesain agar dapat dilakukan pengujian usabilitas nantinya, tugas tugas yang akan dituliskan pada tabel 4.15 berikut ini merupakan informasi tugas utama dari alumni, sedangkan untuk detailnya akan dijelaskan dibab selanjutnya pada perancangan tugas pengguna:

Tabel 4.15 Tugas Pengguna

| No | Tugas Pengguna |
|----|--|
| 1 | Alumni diminta menemukan cara untuk mengirimkan pesan kepada alumni lain yang telah terdaftar dalam sistem <i>tracer study</i> . |
| 2 | Alumni diminta untuk menemukan informasi lowongan pekerjaan yang ada pada sistem <i>tracer study</i> . |
| 3 | Alumni diminta untuk menemukan cara untuk memasukkan informasi pekerjaan yang dimiliki oleh alumni |
| 4 | Alumni diminta menemukan cara untuk mengubah data diri yang telah dimasukkan saat proses registrasi alumni. |
| 5 | Alumni diminta menemukan cara untuk menambahkan informasi pekerjaan alumni yang telah disediakan oleh sistem <i>tracer study</i> . |
| 6 | Alumni diminta menemukan cara untuk menambahkan informasi pendidikan yang telah disediakan oleh sistem <i>tracer study</i> . |
| 7 | Alumni diminta menemukan cara untuk melihat informasi pekerjaan yang telah diisikan oleh alumni. |
| 8 | Alumni diminta menemukan cara untuk melihat informasi pendidikan yang telah diisikan oleh alumni. |

4.8 Analisis Pertanyaan *Tracer Study*

Data pertanyaan yang dapat ditanyakan pada pertanyaan pendidikan alumni dan data pekerjaan alumni dapat dilihat pada lampiran B tentang pertanyaan yang dibutuhkan dalam proses pengambilan data untuk *tracer study*. Data

tersebut dapat berupa data pribadi, data pekerjaan dan pendidikan alumni. Terdapat beberapa perubahan yang diminta oleh ketua program studi mengingat kemudahan akan pengisian pada form aplikasi untuk menggunakan `radio button` daripada `textfield` untuk pertanyaan yang bersifat range dan lebih mudah menggunakan `radio button` untuk pertanyaan yang memiliki jawaban yang sudah pasti seperti program studi. Pertanyaan yang ditampilkan adalah pertanyaan yang telah divalidasi oleh ketua program studi Sistem Informasi, serta pertanyaan yang didapatkan dari hasil wawancara yang terlampir di Lampiran A.



BAB 5 PERANCANGAN

Bab ini menjelaskan tentang tahap yang mengubah rekayasa kebutuhan menjadi sebuah rancangan pengembangan perangkat lunak, yang mana pada tahap ini akan dibuat *activity diagram*, perancangan kode, perancangan data, perancangan antarmuka yang terdiri dari *screenflow diagram* dan perancangan antarmuka itu sendiri, serta perancangan tugas pengguna. Perancangan sistem menggunakan metode *prototyping* yang mana hanya prototype terakhir yang dituliskan dalam skripsi ini.

5.1 Perancangan Arsitektur Sistem

Sistem tracer study membantu alumni dalam proses saling terhubungnya alumni, sistem ini dibangun dengan menggunakan pendekatan Model View Presenter (MVP). MVP adalah salah satu jenis pola perancangan aplikasi yang mana membedakan fungsi dari tampilan dengan fungsional yang berjalan dibelakang aplikasi. Dengan menggunakan MVP diharapkan dapat membuat alur program yang dapat lebih mudah membantu pembaca ataupun pengembang lainnya memahami alur hubungan antara tampilan, pengontrol dan entitas yang ada.



Gambar 5.1 Gambaran Umum Sistem

Dari gambar diatas menjelaskan pertukaran data system yang akan dibangun dengan backend dimana nantinya akan mengatur hak akses *tracer study*. Data yang dikirimkan oleh system akan disimpan dalam *realtime database* yang dimiliki oleh *cloud computing* dari sistem yang disebut sebagai Firebase. Firebase memiliki beberapa fungsi selain sebagai media penyimpanan data juga sebagai

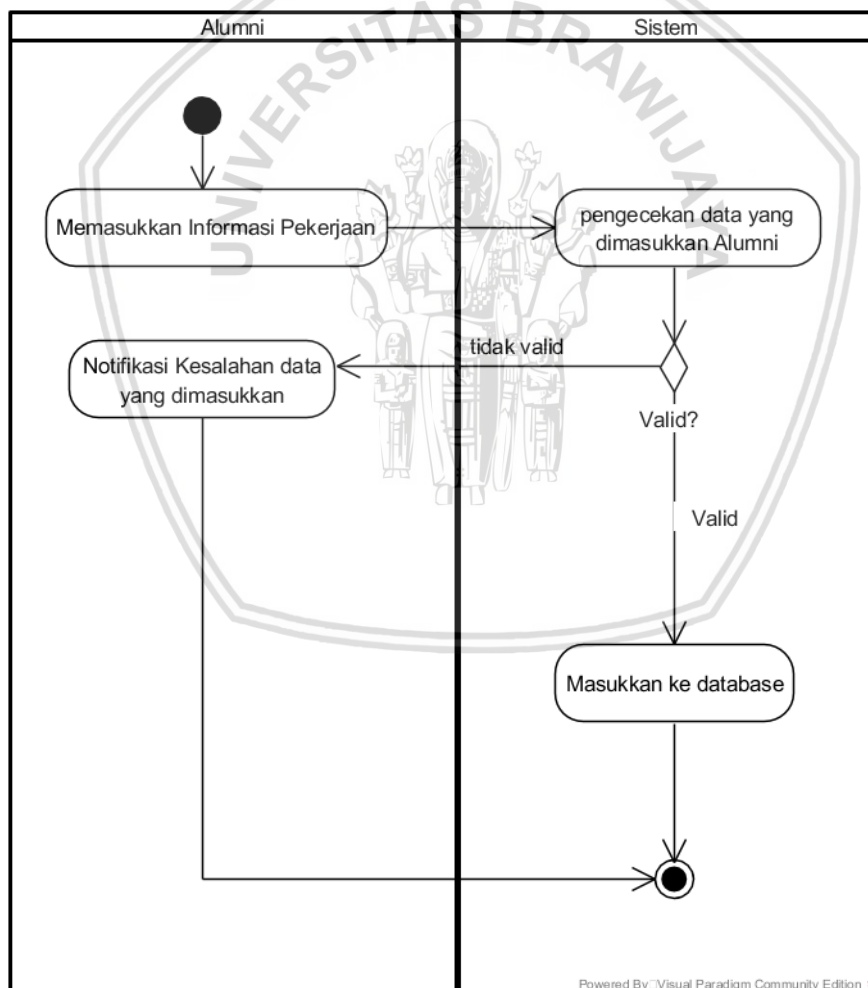
media *authentication*, *cloud storage*. Firebase menggunakan JSON sebagai media pertukaran data.

Sistem tracer study dibuat dengan bahasa pemrograman Java dan XML. Javamerupakan bagian dari bahasa *native* dari android yang mengatur *logic* atau segala sesuatu yang berhubungan dengan fungsi-fungsi sistem. Sedangkan XML adalah bahasa yang digunakan untuk membuat tampilan yang nantinya akan dihubungkan dengan *logic* menggunakan Java. Sistem tracer study menggunakan beberapa sensor, seperti kamera, internet, dan GPS.

5.2 Diagram Activity

Berikut ini adalah *diagram activity* dari sistem *tracer study*, diagram yang dituliskan hanya sebagian dari total 12 kebutuhan yang ada, yaitu tiga *diagram activity* saja.

5.2.1 Masukkan Informasi Pekerjaan

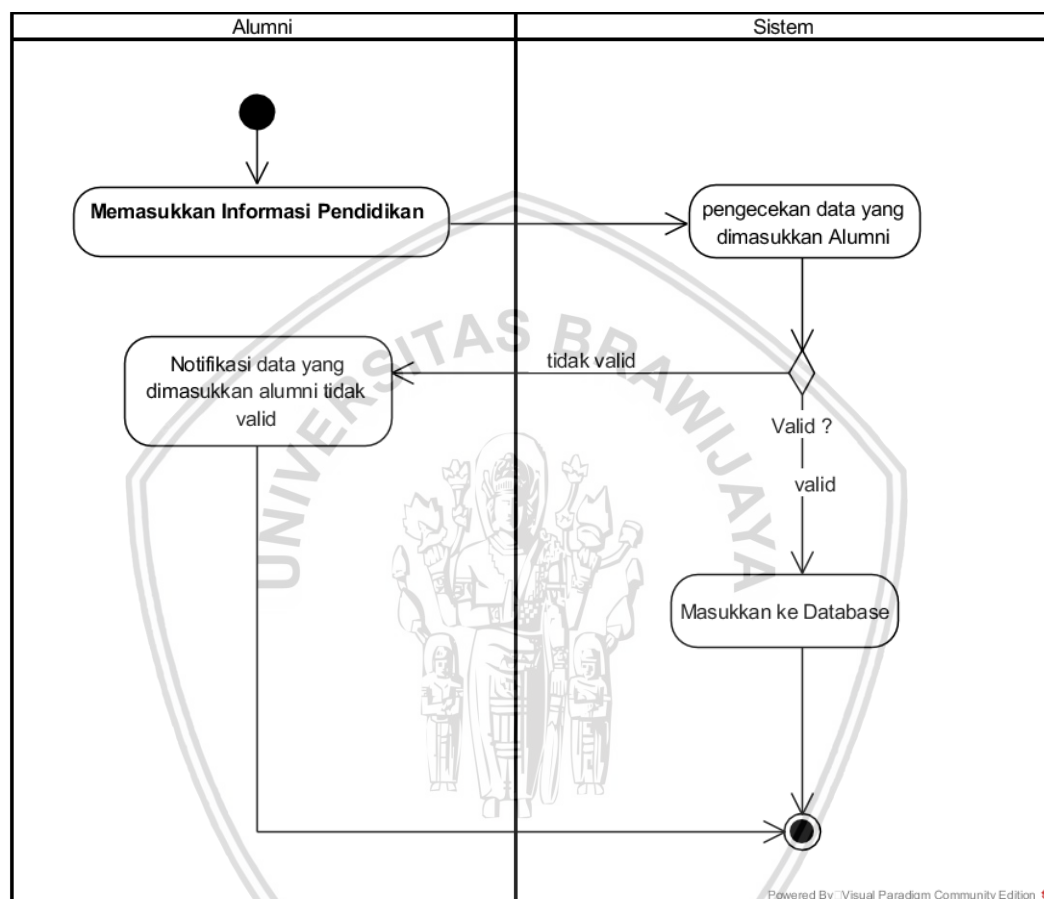


Gambar 5.2 Masukkan Informasi Pekerjaan

Dalam gambar 5.1 menjelaskan tentang bagaimana perilaku aktifitas yang dilakukan oleh aktor dan aktifitas yang dilakukan oleh sistem. Dimana aktor

alumni dapat mengisi informasi pekerjaan yang mana *formnya* telah disediakan oleh sistem, lalu masukan dari alumni akan divalidasi oleh sistem, setelah validasi oleh sistem selesai, jika masukan dari alumni valid, maka sistem akan memasukkan masukan alumni kedalam *database*, lalu aktifitas sistem akan selesai, namun jika masukan dari alumni tidak valid maka sistem akan menampilkan informasi bahwa masukan yang diberikan oleh alumni tidak valid.

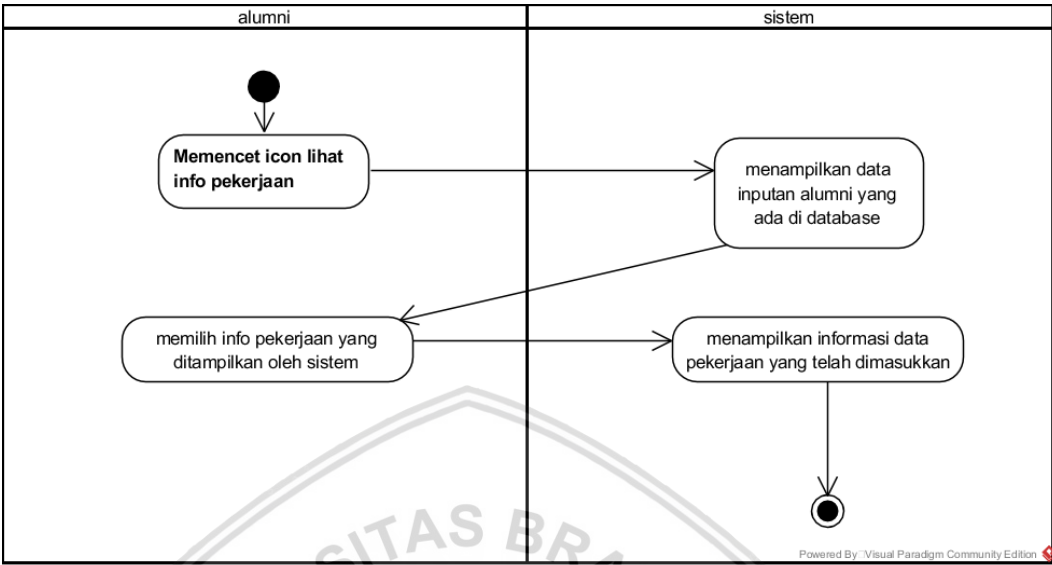
5.2.2 Masukkan Informasi Pendidikan



Gambar 5.3 Memasukkan Informasi Pendidikan

Dalam gambar 5.2 menjelaskan tentang bagaimana perilaku aktifitas yang dilakukan oleh aktor dan aktifitas yang dilakukan oleh sistem. Dimana aktor alumni dapat mengisi informasi pendidikan yang mana *formnya* telah disediakan oleh sistem, lalu masukan dari alumni akan divalidasi oleh sistem, setelah validasi oleh sistem selesai, jika masukan dari alumni valid, maka sistem akan memasukkan masukan alumni kedalam *database*, lalu aktifitas sistem akan selesai, namun jika masukan dari alumni tidak valid maka sistem akan menampilkan informasi bahwa masukan yang diberikan oleh alumni tidak valid.

5.2.3 Melihat Data Pekerjaan



Gambar 5.4 Melihat Data Pekerjaan

Dalam gambar 5.3 menjelaskan tentang bagaimana perilaku aktifitas antara aktor alumni dan sistem, dimana aktor alumni dapat memilih untuk menekan ikon lihat informasi pekerjaan dan sistem menampilkan masukan alumni yang ada di *database*, lalu setelah itu alumni dapat memilih informasi pekerjaan yang ditampilkan oleh sistem, dan sistem akan menampilkan data pekerjaan yang ada di *database* kepada alumni.

5.3 Perancangan Data

Struktur JSON yang berisi tentang pertukaran data antara *firebase realtime database* dan sistem yang sedang akan dibangun. Pertukaran data ini akan berbentuk skema JSON yang nantinya akan diimplementasikan dalam struktur data *noSQL realtime database firebase*. Berikut perancangan data yang terdapat dalam tabel 5.1:

Tabel 5.1 Perancangan Data

| | |
|----|--|
| 1 | { |
| 2 | "\$schema": "http://json-schema.org/draft-04/schema#", |
| 3 | "type": "object", |
| 4 | "properties": { |
| 5 | "TRACER STUDY": { |
| 6 | "type": "object", |
| 7 | "properties": { |
| 8 | "UNIVERSITAS BRAWIJAYA": { |
| 9 | "type": "object", |
| 10 | "properties": { |
| 11 | "FAKULTAS ILMU KOMPUTER": { |
| 12 | "type": "object", |
| 13 | "properties": { |
| 14 | "CHAT": { |

| | |
|----|-------------------|
| 15 | "type": "object", |
| 16 | "properties": { |
| 17 | "CHAT_LIST": { |
| 18 | "type": "object", |
| 19 | "properties": { |
| 20 | "ALUMNI": { |
| 21 | "type": "object", |
| 22 | "properties": { |
| 23 | "CHAT": { |
| 24 | "type": "object", |
| 25 | "properties": { |
| 26 | "string": { |
| 27 | "type": "object", |
| 28 | "properties": { |
| 29 | "string": { |
| 30 | "type": "object", |
| 31 | "properties": { |
| 32 | "imageProfile": { |
| 33 | "type": "string" |
| 34 | }, |
| 35 | "message": { |
| 36 | "type": "string" |
| 37 | }, |
| 38 | "receiverNIM": { |
| 39 | "type": "string" |
| 40 | }, |
| 41 | "receiverName": { |
| 42 | "type": "string" |
| 43 | }, |
| 44 | "senderNIM": { |
| 45 | "type": "string" |
| 46 | }, |
| 47 | "senderName": { |
| 48 | "type": "string" |
| 49 | }, |
| 50 | "timeStamp": { |
| 51 | "type": "string" |
| 52 | } |
| 53 | }, |
| 54 | "required": [|
| 55 | "imageProfile", |
| 56 | "message", |
| 57 | "receiverNIM", |
| 58 | "receiverName", |
| 59 | "senderNIM", |
| 60 | "senderName", |
| 61 | "timeStamp" |
| 62 |] |
| 63 | } |
| 64 | }, |
| 65 | "required": [|
| 66 | "string" |
| 67 |] |
| 68 | } |
| 69 | }, |
| 70 | "required": [|
| 71 | "string" |
| 72 |] |
| 73 | } |

```

74         },
75         "required": [
76             "CHAT"
77         ]
78     },
79 },
80 "required": [
81     "ALUMNI"
82 ]
83 },
84 "USER": {
85     "type": "object",
86     "properties": {
87         "ALUMNI": {
88             "type": "object",
89             "properties": {
90                 "string": {
91                     "type": "object",
92                     "properties": {
93                         "string": {
94                             "type": "object",
95                             "properties": {
96                                 "fileAttach": {
97                                     "type": "string"
98                                 },
99                                 "imageProfile": {
100                                     "type": "string"
101                                 },
102                                 "message": {
103                                     "type": "string"
104                                 },
105                                 "receiverNIM": {
106                                     "type": "string"
107                                 },
108                                 "receiverName": {
109                                     "type": "string"
110                                 },
111                                 "senderNIM": {
112                                     "type": "string"
113                                 },
114                                 "senderName": {
115                                     "type": "string"
116                                 },
117                                 "timeStamp": {
118                                     "type": "string"
119                                 }
120                             },
121                         "required": [
122                             "fileAttach",
123                             "imageProfile",
124                             "message",
125                             "receiverNIM",
126                             "receiverName",
127                             "senderNIM",
128                             "senderName",
129                             "timeStamp"
130                         ]
131                     }
132                 },

```

```

133         "required": [
134             "string"
135         ]
136     },
137 },
138     "required": [
139         "string"
140     ]
141 },
142 },
143     "required": [
144         "ALUMNI"
145     ]
146 },
147 },
148     "required": [
149         "CHAT_LIST",
150         "USER"
151     ]
152 },
153     "DATA": {
154         "type": "object",
155         "properties": {
156             "USER": {
157                 "type": "object",
158                 "properties": {
159                     "ALUMNI": {
160                         "type": "object",
161                         "properties": {
162                             "NIM": {
163                                 "type": "object",
164                                 "properties": {
165                                     "145150200111170": {
166                                         "type": "object",
167                                         "properties": {
168                                             "USER": {
169                                                 "type": "object",
170                                                 "properties": {
171                                                     "alamat": {
172                                                         "type": "string"
173                                                     },
174                                                     "bulan_lulus": {
175                                                         "type": "string"
176                                                     },
177                                                     "created_at": {
178                                                         "type": "string"
179                                                     },
180                                                     "display_name": {
181                                                         "type": "string"
182                                                     },
183                                                     "email": {
184                                                         "type": "string"
185                                                     },
186                                                     "firebaseUID": {
187                                                         "type": "string"
188                                                     },
189                                                     "id": {
190                                                         "type": "integer"
191                                                     },

```

| | |
|-----|--------------------|
| 192 | "jenis_kelamin": { |
| 193 | "type": "string" |
| 194 | }, |
| 195 | "latitude": { |
| 196 | "type": "string" |
| 197 | }, |
| 198 | "longitude": { |
| 199 | "type": "string" |
| 200 | }, |
| 201 | "name": { |
| 202 | "type": "string" |
| 203 | }, |
| 204 | "nim": { |
| 205 | "type": "string" |
| 206 | }, |
| 207 | "no_telp": { |
| 208 | "type": "string" |
| 209 | }, |
| 210 | "photo": { |
| 211 | "type": "string" |
| 212 | }, |
| 213 | "program_studi": { |
| 214 | "type": "string" |
| 215 | }, |
| 216 | "tahun_lulus": { |
| 217 | "type": "string" |
| 218 | }, |
| 219 | "tahun_masuk": { |
| 220 | "type": "string" |
| 221 | } |
| 222 | }, |
| 223 | "required": [|
| 224 | "alamat", |
| 225 | "bulan_lulus", |
| 226 | "created_at", |
| 227 | "display_name", |
| 228 | "email", |
| 229 | "firebaseUID", |
| 230 | "id", |
| 231 | "jenis_kelamin", |
| 232 | "latitude", |
| 233 | "longitude", |
| 234 | "name", |
| 235 | "nim", |
| 236 | "no_telp", |
| 237 | "photo", |
| 238 | "program_studi", |
| 239 | "tahun_lulus", |
| 240 | "tahun_masuk" |
| 241 |] |
| 242 | } |
| 243 | }, |
| 244 | "required": [|
| 245 | "USER" |
| 246 |] |
| 247 | }, |
| 248 | "UID": { |
| 249 | "type": "object", |
| 250 | "properties": { |

| | | |
|-----|-----------|-------------------|
| 251 | | "string": { |
| 252 | | "type": "object", |
| 253 | | "properties": { |
| 254 | | "USER": { |
| 255 | | "type": "object", |
| 256 | | "properties": { |
| 257 | | "alamat": { |
| 258 | | "type": |
| 259 | "string" | |
| 260 | | }, |
| 261 | | "bulan_lulus": { |
| 262 | | "type": |
| 263 | "string" | |
| 264 | | }, |
| 265 | | "created_at": { |
| 266 | | "type": |
| 267 | "string" | |
| 268 | | }, |
| 269 | | "display_name": |
| 270 | { | |
| 271 | | "type": |
| 272 | "string" | |
| 273 | | }, |
| 274 | | "email": { |
| 275 | | "type": |
| 276 | "string" | |
| 277 | | }, |
| 278 | | "firebaseUID": { |
| 279 | | "type": |
| 280 | "string" | |
| 281 | | }, |
| 282 | | "id": { |
| 283 | | "type": |
| 284 | "integer" | |
| 285 | | }, |
| 286 | | "jenis_kelamin": |
| 287 | { | |
| 288 | | "type": |
| 289 | "string" | |
| 290 | | }, |
| 291 | | "latitude": { |
| 292 | | "type": |
| 293 | "string" | |
| 294 | | }, |
| 295 | | "longitude": { |
| 296 | | "type": |
| 297 | "string" | |
| 298 | | }, |
| 299 | | "name": { |
| 300 | | "type": |
| 301 | "string" | |
| 302 | | }, |
| 303 | | "nim": { |
| 304 | | "type": |
| 305 | "string" | |
| 306 | | }, |
| 307 | | "no_telp": { |
| 308 | | "type": |
| 309 | "string" | |


```

310     },
311     "photo": {
312         "type":
313         "string"
314     },
315     "program_studi": {
316         {
317             "type":
318             "string"
319         },
320         "tahun_lulus": {
321             "type":
322             "string"
323         },
324         "tahun_masuk": {
325             "type":
326             "string"
327         }
328     },
329     "required": [
330         "alamat",
331         "bulan_lulus",
332         "created_at",
333         "display_name",
334         "email",
335         "firebaseUID",
336         "id",
337         "jenis_kelamin",
338         "latitude",
339         "longitude",
340         "name",
341         "nim",
342         "no_telp",
343         "photo",
344         "program_studi",
345         "tahun_lulus",
346         "tahun_masuk"
347     ]
348 },
349 },
350 "required": [
351     "USER"
352 ]
353 },
354 },
355 "required": [
356     "BSUVAT6fMLeHkF6br3Ly4GVvZ5D3"
357 ]
358 },
359 },
360 },
361 "required": [
362     "145150200111170",
363     "UID"
364 ]
365 },
366 "DATA_PEKERJAAN": {
367     "type": "object",
368     "properties": {

```

| | |
|-----|-------------------------|
| 369 | "145150200111170": { |
| 370 | "type": "object", |
| 371 | "properties": { |
| 372 | "string": { |
| 373 | "type": "object", |
| 374 | "properties": { |
| 375 | "bidangPekerjaan": { |
| 376 | "type": "string" |
| 377 | }, |
| 378 | "id": { |
| 379 | "type": "integer" |
| 380 | }, |
| 381 | "jabatan": { |
| 382 | "type": "string" |
| 383 | }, |
| 384 | "masihBekerja": { |
| 385 | "type": "string" |
| 386 | }, |
| 387 | "mulaiBekerja": { |
| 388 | "type": "string" |
| 389 | }, |
| 390 | "namaPerusahaan": { |
| 391 | "type": "string" |
| 392 | }, |
| 393 | "nim": { |
| 394 | "type": "string" |
| 395 | }, |
| 396 | "pekerjaanPertama": |
| 397 | "type": "string" |
| 398 | }, |
| 399 | |
| 400 | |
| 401 | "pendapatanPerbulan": { |
| 402 | "type": "string" |
| 403 | }, |
| 404 | "selesaiBekerja": { |
| 405 | "type": "string" |
| 406 | }, |
| 407 | "sesuaiKeahlian": { |
| 408 | "type": "string" |
| 409 | }, |
| 410 | "statusPekerjaan": { |
| 411 | "type": "string" |
| 412 | } |
| 413 | }, |
| 414 | "required": [|
| 415 | "bidangPekerjaan", |
| 416 | "id", |
| 417 | "jabatan", |
| 418 | "masihBekerja", |
| 419 | "mulaiBekerja", |
| 420 | "namaPerusahaan", |
| 421 | "nim", |
| 422 | "pekerjaanPertama", |
| 423 | |
| 424 | "pendapatanPerbulan", |
| 425 | "selesaiBekerja", |
| 426 | "sesuaiKeahlian", |
| 427 | "statusPekerjaan" |

| | |
|-----|----------------------|
| 428 |] |
| 429 | } |
| 430 | }, |
| 431 | "required": [|
| 432 | "string" |
| 433 |] |
| 434 | } |
| 435 | }, |
| 436 | "required": [|
| 437 | "145150200111170" |
| 438 |] |
| 439 | }, |
| 440 | "DATA_PENDIDIKAN": { |
| 441 | "type": "object", |
| 442 | "properties": { |
| 443 | "145150200111170": { |
| 444 | "type": "object", |
| 445 | "properties": { |
| 446 | "string": { |
| 447 | "type": "object", |
| 448 | "properties": { |
| 449 | "gelar_diperoleh": { |
| 450 | "type": "string" |
| 451 | }, |
| 452 | "id": { |
| 453 | "type": "integer" |
| 454 | }, |
| 455 | "kota": { |
| 456 | "type": "string" |
| 457 | }, |
| 458 | "levelPendidikan": { |
| 459 | "type": "string" |
| 460 | }, |
| 461 | "negara": { |
| 462 | "type": "string" |
| 463 | }, |
| 464 | "nim": { |
| 465 | "type": "string" |
| 466 | }, |
| 467 | "programStudi": { |
| 468 | "type": "string" |
| 469 | }, |
| 470 | "tahun_lulus": { |
| 471 | "type": "string" |
| 472 | }, |
| 473 | "universitas": { |
| 474 | "type": "string" |
| 475 | } |
| 476 | }, |
| 477 | "required": [|
| 478 | "gelar_diperoleh", |
| 479 | "id", |
| 480 | "kota", |
| 481 | "levelPendidikan", |
| 482 | "negara", |
| 483 | "nim", |
| 484 | "programStudi", |
| 485 | "tahun_lulus", |
| 486 | "universitas" |

```

487         ]
488     }
489 },
490     "required": [
491         "string"
492     ]
493 }
494 },
495     "required": [
496         "145150200111170"
497     ]
498 }
499 },
500     "required": [
501         "NIM",
502         "DATA_PEKERJAAN",
503         "DATA_PENDIDIKAN"
504     ]
505 }
506 },
507     "required": [
508         "ALUMNI"
509     ]
510 },
511     "TIMELINE": {
512         "type": "object",
513         "properties": {
514             "USER": {
515                 "type": "object",
516                 "properties": {
517                     "ALUMNI": {
518                         "type": "object",
519                         "properties": {
520                             "string": {
521                                 "type": "object",
522                                 "properties": {
523                                     "alamatPerusahaan": {
524                                         "type": "string"
525                                     },
526                                     "deskripsiPerusahaan": {
527                                         "type": "string"
528                                     },
529                                 }
530                             "detailDeskripsiPekerjaan": {
531                                 "type": "string"
532                             },
533                             "id": {
534                                 "type": "integer"
535                             },
536                             "jenisPekerjaan": {
537                                 "type": "string"
538                             },
539                             "kontak": {
540                                 "type": "string"
541                             },
542                             "latLokasi": {
543                                 "type": "number"
544                             },
545                             "lngLokasi": {

```

```

546         "type": "number"
547     },
548     "logoPerusahaan": {
549         "type": "string"
550     },
551     "namaPerusahaan": {
552         "type": "string"
553     },
554     "statusPekerjaan": {
555         "type": "string"
556     },
557     "syaratPekerjaan": {
558         "type": "string"
559     },
560     "syaratPendaftaran": {
561         "type": "string"
562     },
563     "syaratPengalamanKerja":
564 {
565         "type": "string"
566     },
567     "timelineDeadline": {
568         "type": "integer"
569     },
570     "timelineType": {
571         "type": "integer"
572     }
573 },
574     "required": [
575         "alamatPerusahaan",
576         "deskripsiPerusahaan",
577         "detailDeskripsiPekerjaan",
578         "id",
579         "jenisPekerjaan",
580         "kontak",
581         "latLokasi",
582         "lngLokasi",
583         "logoPerusahaan",
584         "namaPerusahaan",
585         "statusPekerjaan",
586         "syaratPekerjaan",
587         "syaratPendaftaran",
588         "syaratPengalamanKerja",
589         "timelineDeadline",
590         "timelineType"
591     ]
592 },
593     "string": {
594         "type": "object",
595         "properties": {
596             "alamatPerusahaan": {
597                 "type": "string"
598             },
599             "deskripsiPerusahaan": {
600                 "type": "string"
601             },
602             "detailDeskripsiPekerjaan": {

```

| | | |
|-----|-----------------------------|--------------------------|
| 605 | | "type": "string" |
| 606 | | }, |
| 607 | | "id": { |
| 608 | | "type": "integer" |
| 609 | | }, |
| 610 | | "jenisPekerjaan": { |
| 611 | | "type": "string" |
| 612 | | }, |
| 613 | | "kontak": { |
| 614 | | "type": "string" |
| 615 | | }, |
| 616 | | "latLokasi": { |
| 617 | | "type": "number" |
| 618 | | }, |
| 619 | | "lngLokasi": { |
| 620 | | "type": "number" |
| 621 | | }, |
| 622 | | "logoPerusahaan": { |
| 623 | | "type": "string" |
| 624 | | }, |
| 625 | | "namaPerusahaan": { |
| 626 | | "type": "string" |
| 627 | | }, |
| 628 | | "statusPekerjaan": { |
| 629 | | "type": "string" |
| 630 | | }, |
| 631 | | "syaratPekerjaan": { |
| 632 | | "type": "string" |
| 633 | | }, |
| 634 | | "syaratPendaftaran": { |
| 635 | | "type": "string" |
| 636 | | }, |
| 637 | | "syaratPengalamanKerja": |
| 638 | { | |
| 639 | | "type": "string" |
| 640 | | }, |
| 641 | | "timelineDeadline": { |
| 642 | | "type": "integer" |
| 643 | | }, |
| 644 | | "timelineType": { |
| 645 | | "type": "integer" |
| 646 | | } |
| 647 | | }, |
| 648 | | "required": [|
| 649 | | "alamatPerusahaan", |
| 650 | | "deskripsiPerusahaan", |
| 651 | | |
| 652 | "detailDeskripsiPekerjaan", | |
| 653 | | "id", |
| 654 | | "jenisPekerjaan", |
| 655 | | "kontak", |
| 656 | | "latLokasi", |
| 657 | | "lngLokasi", |
| 658 | | "logoPerusahaan", |
| 659 | | "namaPerusahaan", |
| 660 | | "statusPekerjaan", |
| 661 | | "syaratPekerjaan", |
| 662 | | "syaratPendaftaran", |
| 663 | | "syaratPengalamanKerja", |

| | |
|-----|--------------------------|
| 664 | "timelineDeadline", |
| 665 | "timelineType" |
| 666 |] |
| 667 | } |
| 668 | }, |
| 669 | "required": [|
| 670 | "string", |
| 671 | "string" |
| 672 |] |
| 673 | } |
| 674 | }, |
| 675 | "required": [|
| 676 | "ALUMNI" |
| 677 |] |
| 678 | } |
| 679 | }, |
| 680 | "required": [|
| 681 | "USER" |
| 682 |] |
| 683 | } |
| 684 | }, |
| 685 | "required": [|
| 686 | "USER", |
| 687 | "TIMELINE" |
| 688 |] |
| 689 | } |
| 690 | }, |
| 691 | "required": [|
| 692 | "CHAT", |
| 693 | "DATA" |
| 694 |] |
| 695 | } |
| 696 | }, |
| 697 | "required": [|
| 698 | "FAKULTAS ILMU KOMPUTER" |
| 699 |] |
| 700 | } |
| 701 | }, |
| 702 | "required": [|
| 703 | "UNIVERSITAS BRAWIJAYA" |
| 704 |] |
| 705 | } |
| 706 | }, |
| 707 | "required": [|
| 708 | "TRACER STUDY" |
| 709 |] |
| 710 | } |

Seperti yang terlihat pada tabel 5.1 diatas, terbagi menjadi Nama Aplikasi, Nama Universitas, Nama Fakultas, lalu dibagi menjadi 3 bagian, yaitu *chat*, *data* dan *timeline*, dimana *chat* memiliki anak *user*, lalu memiliki anak *alumni* dan *group*, dimana isi dari anak *alumni* ini adalah pesan antara sesama *alumni* yang melakukan pengiriman pesan pribadi, sedangkan untuk *group* berisi pengiriman pesan yang sifatnya bukan pesan pribadi, namun dapat diisi oleh semua elemen *alumni*, yang kemudian setiap anak akan memiliki *push id* yang mana digunakan sebagai pembeda setiap pesan yang dikirim.

Untuk anak dari *timeline* adalah *user*, lalu *alumni* dan *push id* yang merupakan pembeda dari setiap *timeline*, dimana *timeline* akan dihubungkan dengan aktor *alumni* dan *admin*, sehingga ketika ada perubahan maka dapat langsung berubah di sisi pengguna *alumni* dan *admin*. *timeline* akan menyimpan data untuk lowongan pekerjaan *alumni*.

5.4 Perancangan Antarmuka

Perancangan ini akan membuat *screenflow diagram* dari aplikasi, yang akan dilanjutkan dengan perancangan antarmuka itu sendiri, lalu dilanjutkan dengan perancangan tugas pengguna.

5.4.1 Screenflow Diagram

Dalam gambar 5.4 dijelaskan tentang urutan urutan dari tampilan yang ada di sistem *tracer study*, halaman yang ditampilkan adalah beberapa halaman utama yang ada dalam sistem *tracer study*. Penjelasan lebih lanjut tentang urutan aksi yang didapatkan akan dijelaskan dalam tabel 5.2 sesuai nomor yang ada dalam gambar 5.4 berikut:

Tabel 5.2 Informasi Screenflow Tracer Study

| No | Keterangan |
|----|--|
| 1 | Setelah melakukan <i>login / sign in</i> dan berhasil melakukan otentikasi maka <i>alumni</i> akan diarahkan ke halaman pesan. |
| 2 | Setelah melakukan <i>register</i> dan berhasil melakukan otentikasi maka <i>alumni</i> akan diarahkan ke halaman pesan. |
| 3 | Saat <i>alumni</i> menekan <i>tab</i> LoKer maka <i>alumni</i> akan diarahkan ke halaman lowongan pekerjaan. |
| 4 | Saat <i>alumni</i> menekan <i>tab</i> profil maka <i>alumni</i> akan diarahkan ke halaman profil. |
| 5 | Saat <i>alumni</i> menekan <i>tab chat</i> maka <i>alumni</i> akan diarahkan ke halaman pesan. |

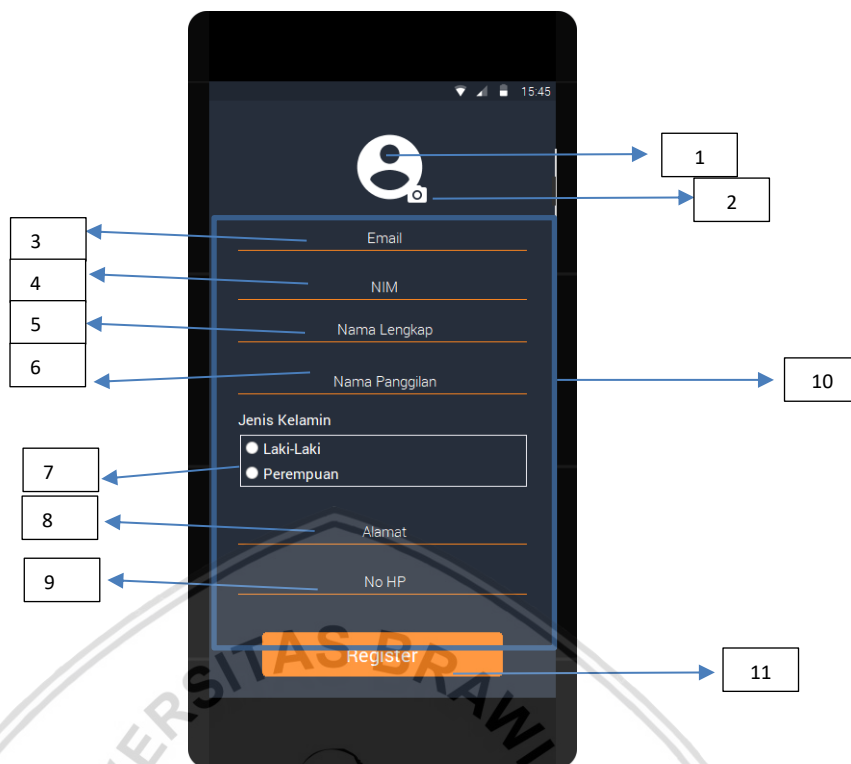


Gambar 5.5 Screenflow *Tracer Study*

5.4.2 Perancangan Antarmuka

Perancangan antarmuka akan menjelaskan detail dari setiap halaman utama aplikasi tracer study, dan dijelaskan dalam tabel yang menjelaskan detail tiap fungsi dari penjelasan antarmuka.

5.4.2.1 Halaman *Register*



Gambar 5.6 Halaman *Register*

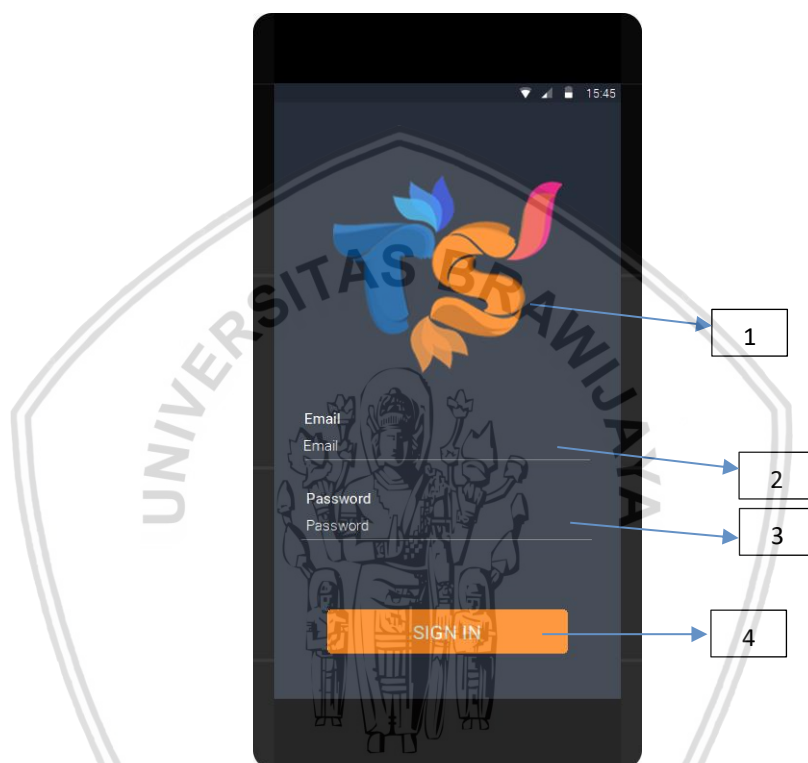
Dalam gambar 5.5 dijelaskan antarmuka halaman *register* yang mana tiap elemen akan dijelaskan dalam tabel 5.3 berikut:

Tabel 5.3 Informasi Elemen Halaman *Register*

| No | Penjelasan | Detail Fungsi |
|----|------------------------|--|
| 1 | Gambar profil pengguna | Digunakan untuk menampilkan foto alumni atau mengubah foto alumni |
| 2 | Ikon foto | Digunakan untuk menerangkan bahwa gambar profil pengguna dapat diubah |
| 3 | Masukan <i>Email</i> | Digunakan untuk memasukkan <i>email</i> dengan format masukan <i>email</i> |
| 4 | Masukan NIM | Digunakan untuk memasukkan nim alumni |
| 5 | Masukan Nama Lengkap | Digunakan untuk memasukkan nama lengkap alumni |
| 6 | Masukan Nama Panggilan | Digunakan untuk memasukkan nama panggilan alumni |
| 7 | Masukan Jenis Kelamin | Digunakan untuk memilih jenis kelamin alumni |
| 8 | Masukan Alamat | Digunakan untuk memasukkan alamat alumni |

| | | |
|----|------------------------|--|
| 9 | Masukan No HP | Digunakan untuk memasukkan no hp alumni |
| 10 | Masukan Data | Digunakan untuk memasukkan masukan data sesuai pertanyaan yang diberikan |
| 11 | <i>Button Register</i> | Digunakan untuk melakukan proses registrasi alumni |

5.4.2.2 Halaman Login



Gambar 5.7 Halaman Login

Dalam gambar 5.6 dijelaskan antarmuka halaman *login* yang mana tiap elemen akan dijelaskan dalam tabel 5.4 berikut:

Tabel 5.4 Informasi Elemen Halaman Login

| No | Penjelasan | Detail Fungsi |
|----|--------------------------|---|
| 1 | Logo <i>tracer study</i> | Digunakan untuk memperjelas aplikasi yang sedang dibuka oleh alumni |
| 2 | Masukan <i>Email</i> | Digunakan untuk memasukkan masukan <i>email</i> terdaftar alumni |
| 3 | Masukan <i>Password</i> | Digunakan untuk memasukkan masukan <i>password</i> alumni |

| | | |
|---|----------------|---|
| 4 | Tombol Sign In | digunakan untuk mengawali proses otentikasi |
|---|----------------|---|

5.4.2.3 Halaman Pesan



Gambar 5.8 Halaman *Chat*

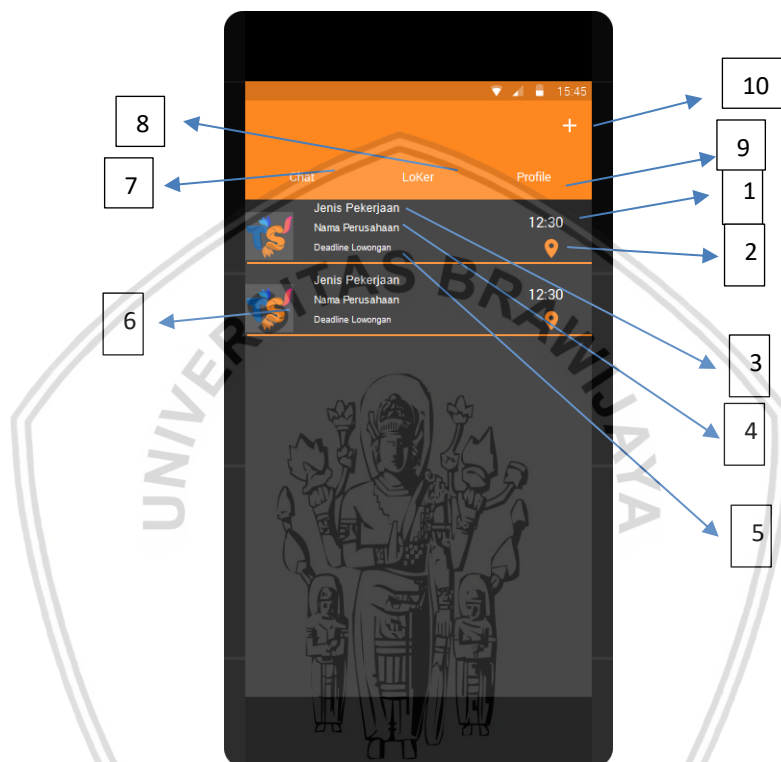
Dalam gambar 5.7 dijelaskan antarmuka halaman *chat* yang mana tiap elemen akan dijelaskan dalam tabel 5.5 berikut:

Tabel 5.5 Informasi Elemen Halaman *Chat*

| No | Penjelasan | Detail Fungsi |
|----|----------------------|--|
| 1 | Waktu Kirim | Digunakan untuk menunjukkan jam pengiriman pesan |
| 2 | Nama | Digunakan untuk menunjukkan nama teman <i>chat</i> dari alumni |
| 3 | <i>Chat</i> terakhir | Digunakan untuk menunjukkan isi pesan terakhir yang diterima oleh alumni |
| 4 | Foto | Digunakan untuk menunjukkan foto profil dari teman alumni |
| 5 | <i>Tab Chat</i> | Digunakan untuk mengarahkan alumni ke halaman <i>chat</i> |

| | | |
|---|-------------|--|
| 6 | Tab LoKer | Digunakan untuk mengarahkan alumni ke halaman Lowongan Pekerjaan |
| 7 | Tab Profil | Digunakan untuk mengarahkan alumni ke halaman profil |
| 8 | Ikon tambah | Digunakan untuk menambahkan teman chat alumni |

5.4.2.4 Halaman Lowongan Pekerjaan



Gambar 5.9 Halaman Lowongan Pekerjaan

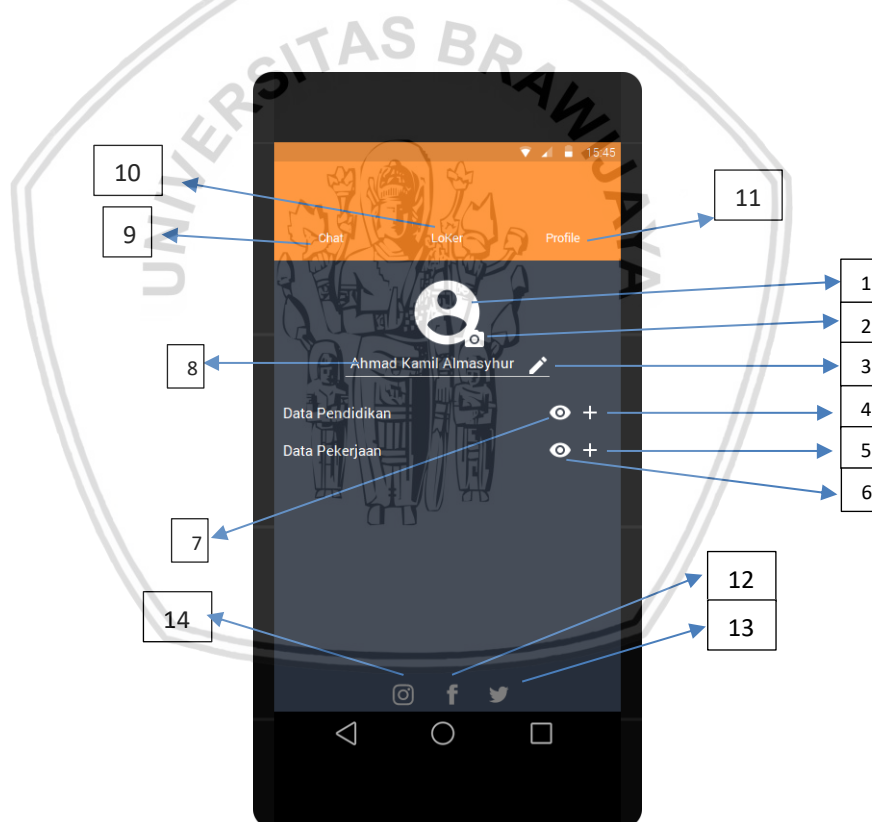
Dalam gambar 5.8 dijelaskan antarmuka halaman lowongan pekerjaan yang mana tiap elemen akan dijelaskan dalam tabel 5.6 berikut:

Tabel 5.6 Informasi Elemen Halaman Lowongan Pekerjaan

| No | Penjelasan | Detail Fungsi |
|----|--------------------------------|--|
| 1 | Batas Waktu Lowongan Pekerjaan | Menjelaskan tentang batas waktu untuk mendaftar di lowongan pekerjaan tersebut |
| 2 | Ikon lokasi | Jika ditap akan menjelaskan tentang lokasi dari lowongan pekerjaan dengan menggunakan maps |
| 3 | Jenis pekerjaan | Menjelaskan tentang jenis pekerjaan yang disediakan pada lowongan pekerjaan |
| 4 | Nama Perusahaan | Menjelaskan tentang nama perusahaan pembuka lowongan pekerjaan |

| | | |
|----|-------------------|---|
| 5 | Deadline Lowongan | Menjelaskan tentang deadline lowongan pekerjaan |
| 6 | Icon Perusahaan | Menjelaskan tentang icon dari perusahaan atau foto dari informasi lowongan pekerjaan tersebut |
| 7 | Tab Chat | Digunakan untuk mengarahkan alumni ke halaman chat |
| 8 | Tab LoKer | Digunakan untuk mengarahkan alumni ke halaman Lowongan Pekerjaan |
| 9 | Tab Profil | Digunakan untuk mengarahkan alumni ke halaman profil |
| 10 | Ikon tambah | Digunakan untuk menambahkan lowongan pekerjaan yang alumni miliki |

5.4.2.5 Halaman Profil



Gambar 5.10 Halaman Profil

Dalam gambar 5.9 menjelaskan antarmuka halaman profil yang mana tiap elemen akan dijelaskan dalam tabel 5.7 berikut:

Tabel 5.7 Informasi Elemen Halaman Profil

| No | Penjelasan | Detail Fungsi |
|----|--------------------|---|
| 1 | Gambar user profil | Digunakan untuk menampilkan foto alumni |

| | | |
|----|-------------------|--|
| 2 | Ikon foto | Digunakan untuk menambahkan atau mengubah foto profil dari alumni |
| 3 | Ikon pensil | Digunakan untuk mengubah informasi data diri dari alumni |
| 4 | Ikon tambah | Digunakan untuk menambahkan data Pendidikan dari alumni |
| 5 | Ikon tambah | Digunakan untuk menambahkan data pekerjaan dari alumni |
| 6 | Ikon Mata | Digunakan untuk melihat data pendidikan dari alumni |
| 7 | Ikon Mata | Digunakan untuk melihat data pekerjaan dari alumni |
| 8 | Nama | Digunakan untuk melihat nama alumni yang sedang masuk kedalam sistem |
| 9 | <i>Tab Chat</i> | Digunakan untuk mengarahkan alumni ke halaman <i>chat</i> |
| 10 | <i>Tab LoKer</i> | Digunakan untuk mengarahkan alumni ke halaman Lowongan Pekerjaan |
| 11 | <i>Tab Profil</i> | Digunakan untuk mengarahkan alumni ke halaman profil |
| 12 | Ikon Instagram | Digunakan untuk mengarahkan alumni ke halaman resmi instagram Filkom |
| 13 | Ikon facebook | Digunakan untuk mengarahkan alumni ke halaman resmi facebook Filkom |
| 14 | Ikon twitter | Digunakan untuk mengarahkan alumni ke halaman resmi twitter Filkom |

5.5 Perancangan Tugas Pengguna

Perancangan tugas pengguna adalah sebuah cara untuk menentukan apa saja yang seharusnya dilakukan oleh pengguna dalam proses penggunaan aplikasi yang dapat dilakukan oleh alumni. Perancangan tugas pengguna diambil dari *Use Case Diagram* dan setidaknya mewakili semua kebutuhan yang ada didalam Use Case. untuk Dalam bagian ini, perancangan tugas pengguna akan menjelaskan detail tentang tugas-tugas yang akan dilakukan oleh pengguna dalam penggunaan aplikasi. Berikut ini adalah tugas-tugas yang dapat dilakukan oleh pengguna dalam proses penggunaan aplikasi *tracer study*:

- Alumni diminta untuk menemukan cara untuk mengirim pesan kepada alumni lain pada sistem *tracer study*

1. Alumni diminta untuk menambahkan teman yang dikenal oleh alumni pada saat proses studi di FILKOM.
2. Alumni diminta untuk mengetikkan pesan yang ditujukan untuk teman alumni.
3. Alumni diminta untuk mengirimkan pesan.
- b. Alumni diminta untuk menemukan informasi lowongan pekerjaan dalam sistem *tracer study*
 1. Alumni diminta untuk mencari bagian tampilan khusus untuk lowongan pekerjaan
 2. Alumni diminta untuk mencari salah satu lowongan pekerjaan yang diinginkan dan melihat detailnya
- c. Alumni diminta untuk menemukan cara untuk memasukkan informasi pekerjaan yang dimiliki oleh alumni
 1. Alumni diminta untuk mencari bagian tampilan khusus untuk lowongan pekerjaan
 2. Alumni diminta untuk mencari tombol tambahkan pekerjaan
 3. Alumni diminta untuk menambahkan lowongan pekerjaan
- d. Alumni diminta untuk menemukan cara untuk mengubah data diri yang telah dimasukkan saat *register*
 1. Alumni diminta untuk mencari halaman profil
 2. Alumni diminta untuk mencari bagian untuk mengubah data diri alumni
 3. Alumni diminta untuk mengubah data diri alumni
 4. Alumni diminta untuk menyimpan perubahan data diri alumni
- e. Alumni diminta untuk menemukan cara untuk menambahkan informasi pekerjaan alumni yang ada di sistem *tracer study*
 1. Alumni diminta untuk mencari profil alumni
 2. Alumni diminta untuk mencari cara untuk menambahkan data pekerjaan
 3. Alumni diminta untuk mengisi data yang dibutuhkan pada data pekerjaan
 4. Alumni diminta untuk menyimpan data pekerjaan yang sudah diisi
- f. Alumni diminta untuk menemukan cara untuk melihat informasi pekerjaan yang telah diisi oleh alumni
 1. Alumni diminta untuk mencari profil alumni
 2. Alumni diminta untuk mencari cara untuk melihat informasi data pekerjaan alumni
 3. Alumni diminta untuk melihat detail data pekerjaan yang telah alumni isikan

- g. Alumni diminta untuk menemukan cara untuk menambahkan informasi pendidikan alumni yang ada di sistem *tracer study*
 - 1. Alumni diminta untuk mencari profil alumni
 - 2. Alumni diminta untuk mencari cara untuk menambahkan data pendidikan
 - 3. Alumni diminta untuk mengisi data yang dibutuhkan pada data pendidikan
- h. Alumni diminta untuk menemukan cara untuk melihat informasi pendidikan yang telah diisi oleh alumni
 - 1. Alumni diminta untuk mencari profil alumni
 - 2. Alumni diminta untuk mencari cara untuk melihat informasi data pendidikan alumni
 - 3. Alumni diminta untuk melihat detail data pendidikan yang telah alumni isikan



BAB 6 IMPLEMENTASI

Bab ini akan menjelaskan tentang implementasi *activity diagram* menjadi kode program, rancangan data menjadi implementasi data, serta rancangan *user interface* menjadi implementasi *user interface*.

6.1 Implementasi

Pada tahap ini akan dijelaskan tentang proses implementasi yang ada di sistem *tracer study* yang meliputi implementasi kode program, implementasi data, serta implementasi antarmuka. Dimana implementasi ini merupakan bentuk kelanjutan dari proses analisis dan perancangan yang telah dilakukan sebelumnya.

6.1.1 Implementasi Kode Program

6.1.1.1 Implementasi Kode Program Masukkan Informasi Pekerjaan

Tabel 6.1 Kode Masukkan Informasi Pekerjaan

| No | Kode |
|----|---|
| 1 | <code>private int requestCode = 0;</code> |
| 2 | <code>public static final int REQUEST_MULAI = 198;</code> |
| 3 | <code>public static final int REQUEST_SELESAI = 199;</code> |
| 4 | <code>DataPekerjaan dataPekerjaan;</code> |
| 5 | |
| 6 | <code>@Override</code> |
| 7 | <code>protected void onCreate(Bundle savedInstanceState) {</code> |
| 8 | <code> super.onCreate(savedInstanceState);</code> |
| 9 | <code> setContentView(R.layout.activity_add_info_pekerjaan);</code> |
| 10 | <code> ButterKnife.bind(this);</code> |
| 11 | <code> setSupportActionBar(toolbar);</code> |
| 12 | <code> getSupportActionBar().setDisplayHomeAsUpEnabled(true);</code> |
| 13 | <code>}</code> |
| 14 | |
| 15 | <code>@OnClick(R.id.btn_info_pekerjaan_save_record)</code> |
| 16 | <code>void onSaveRecordClicked(View view) {</code> |
| 17 | <code> if (validateCurrentPage()) {</code> |
| 18 | <code> Toast.makeText(this, "Data Berhasil</code> |
| 19 | <code>Disimpan\nTerima Kasih", Toast.LENGTH_SHORT).show();</code> |
| 20 | <code> finish();</code> |
| 21 | <code> }</code> |
| 22 | <code>}</code> |
| 23 | |
| 24 | <code>@Override</code> |
| 25 | <code>public boolean onOptionsItemSelected(MenuItem item) {</code> |
| 26 | <code> switch (item.getItemId()) {</code> |
| 27 | <code> case android.R.id.home:</code> |
| 28 | <code> onBackPressed();</code> |
| 29 | <code> return true;</code> |
| 30 | <code> }</code> |
| 31 | <code> return super.onOptionsItemSelected(item);</code> |
| 32 | <code>}</code> |
| 33 | |
| 34 | <code>@OnClick(R.id.btn_info_pekerjaan_get_via_calendar_mulai)</code> |


```

35     void onGetViaCalendarMulaiClicked(View view) {
36         DatePickerFragment datePickerFragment = new
37         DatePickerFragment (AddInfoPekerjaanActivity.this);
38         datePickerFragment.show (getFragmentManager(),
39         "TYPE_DATE");
40         requestCode = REQUEST_MULAI;
41     }
42
43
44     @OnClick(R.id.btn_info_pekerjaan_get_via_calendar_selesai)
45     void onGetViaCalendarSelesaiClicked(View view) {
46         DatePickerFragment datePickerFragment = new
47         DatePickerFragment (AddInfoPekerjaanActivity.this);
48         datePickerFragment.show (getFragmentManager(),
49         "TYPE_DATE");
50         requestCode = REQUEST_SELESAI;
51     }
52     private boolean validateCurrentPage() {
53         String mulaiBekerjaAlumni =
54         mulaiBekerja.getText().toString();
55         String selesaiBekerjaAlumni =
56         selesaiBekerja.getText().toString();
57         boolean masihBekerjaAlumni =
58         masihBekerja.isChecked();
59         boolean pekerjaanPertamaAlumni =
60         pekerjaanPertama.isChecked();
61         RadioButton statusPekerjaanAlumn =
62         findViewById(statusPekerjaan.getCheckedRadioButtonId());
63         if (null == statusPekerjaanAlumn) {
64             Snackbar.make (findViewById (android.R.id.content),
65             getString(R.string.is_required,
66             getString(R.string.status_pekerjaan)),
67             Snackbar.LENGTH_SHORT).show();
68             return false;
69         }
70         String statusPekerjaanAlumni =
71         statusPekerjaanAlumn.getText().toString();
72         String bidangPekerjaanAlumni =
73         bidangPekerjaan.getText().toString();
74         RadioButton sesuaiKeahlianAlumn =
75         findViewById(sesuaiKeahlian.getCheckedRadioButtonId());
76         String sesuaiKeahlianAlumni =
77         sesuaiKeahlianAlumn.getText().toString();
78         String namaPerusahaanAlumni =
79         namaPerusahaan.getText().toString();
80         String jabatanAlumni = jabatan.getText().toString();
81         RadioButton gajiPerbulanAlumn =
82         findViewById(gajiPerbulan.getCheckedRadioButtonId());
83         String gajiPerbulanAlumni =
84         gajiPerbulanAlumn.getText().toString();
85
86         if (!Tools.isThisStringValid(mulaiBekerjaAlumni)) {
87             Toast.makeText (this,
88             getString(R.string.is_required,
89             getString(R.string.mulai_bekerja)),
90             Toast.LENGTH_SHORT).show();
91             return false;
92         }
93         if (!masihBekerjaAlumni) {

```

```

94         if
95         (!Tools.isThisStringValid(selesaiBekerjaAlumni)) {
96             Toast.makeText(this,
97                 getString(R.string.is_required,
98                     getString(R.string.selesai_bekerja)),
99                 Toast.LENGTH_SHORT).show();
100             return false;
101         }
102     }
103     if (!Tools.isThisStringValid(statusPekerjaanAlumni))
104     {
105         Toast.makeText(this,
106             getString(R.string.is_required,
107                 getString(R.string.status_pekerjaan)),
108             Toast.LENGTH_SHORT).show();
109         return false;
110     }
111     if (!Tools.isThisStringValid(bidangPekerjaanAlumni))
112     {
113         Toast.makeText(this,
114             getString(R.string.is_required,
115                 getString(R.string.bidang_pekerjaan)),
116             Toast.LENGTH_SHORT).show();
117         return false;
118     }
119     if (!Tools.isThisStringValid(sesuaiKeahlianAlumni)) {
120         Toast.makeText(this,
121             getString(R.string.is_required,
122                 getString(R.string.sesuai_keahlian)),
123             Toast.LENGTH_SHORT).show();
124         return false;
125     }
126     if (!Tools.isThisStringValid(namaPerusahaanAlumni)) {
127         Toast.makeText(this,
128             getString(R.string.is_required,
129                 getString(R.string.nama_perusahaan)),
130             Toast.LENGTH_SHORT).show();
131         return false;
132     }
133     if (!Tools.isThisStringValid(jabatanAlumni)) {
134         Toast.makeText(this,
135             getString(R.string.is_required,
136                 getString(R.string.jabatan)), Toast.LENGTH_SHORT).show();
137         return false;
138     }
139     if (!Tools.isThisStringValid(gajiPerbulanAlumni)) {
140         Toast.makeText(this,
141             getString(R.string.is_required,
142                 getString(R.string.pendapatan_perbulan)),
143             Toast.LENGTH_SHORT).show();
144         return false;
145     }
146
147     if (null == this.dataPekerjaan){
148         DataPekerjaan dataPekerjaan = new
149         DataPekerjaan();
150         dataPekerjaan.setId(Tools.getCurrentTime());
151
152         dataPekerjaan.setNim(GlobalData.getSharedGlobalData().getUse

```

```

153 r().getNim());
154
155 dataPekerjaan.setMulaiBekerja(mulaiBekerjaAlumni);
156
157 dataPekerjaan.setSelesaiBekerja(selesaiBekerjaAlumni);
158 String masihBekerjakahAlumni = (true ==
159 masihBekerjaAlumni ? "1" : "0");
160
161 dataPekerjaan.setMasihBekerja(masihBekerjakahAlumni);
162 String pekerjaanPertamakahAlumni = (true ==
163 pekerjaanPertamaAlumni ? "1" : "0");
164
165 dataPekerjaan.setPekerjaanPertama(pekerjaanPertamakahAlumni)
166 ;
167
168 dataPekerjaan.setStatusPekerjaan(statusPekerjaanAlumni);
169
170 dataPekerjaan.setBidangPekerjaan(bidangPekerjaanAlumni);
171
172 dataPekerjaan.setSesuaiKeahlian(sesuaiKeahlianAlumni);
173
174 dataPekerjaan.setNamaPerusahaan(namaPerusahaanAlumni);
175 dataPekerjaan.setJabatan(jabatanAlumni);
176
177 dataPekerjaan.setPendapatanPerbulan(gajiPerbulanAlumni);
178
179 DatabaseReference reference =
180 FirebaseDatabase.getInstance().getReference();
181 reference.child(Global.TRACER_STUDY)
182 .child(Global.UNIVERSITY)
183 .child(Global.FACULTY)
184 .child(Global.TYPE_DATA)
185 .child(Global.TYPE_USER)
186 .child(Global.TYPE_DATA_PKERJAAN)
187
188 .child(GlobalData.getSharedGlobalData().getUser().getNim())
189 .push()
190 .setValue(dataPekerjaan);
191 } else {
192 DataPekerjaan dataPekerjaan = new
193 DataPekerjaan();
194 dataPekerjaan.setId(this.dataPekerjaan.getId());
195
196 dataPekerjaan.setNim(this.dataPekerjaan.getNim());
197
198 dataPekerjaan.setMulaiBekerja(mulaiBekerjaAlumni);
199
200 dataPekerjaan.setSelesaiBekerja(selesaiBekerjaAlumni);
201 String masihBekerjakahAlumni = (true ==
202 masihBekerjaAlumni ? "1" : "0");
203
204 dataPekerjaan.setMasihBekerja(masihBekerjakahAlumni);
205 String pekerjaanPertamakahAlumni = (true ==
206 pekerjaanPertamaAlumni ? "1" : "0");
207
208 dataPekerjaan.setPekerjaanPertama(pekerjaanPertamakahAlumni)
209 ;
210
211 dataPekerjaan.setStatusPekerjaan(statusPekerjaanAlumni);

```

```

212 dataPekerjaan.setBidangPekerjaan(bidangPekerjaanAlumni);
213
214 dataPekerjaan.setSesuaiKeahlian(sesuaiKeahlianAlumni);
215
216 dataPekerjaan.setNamaPerusahaan(namaPerusahaanAlumni);
217 dataPekerjaan.setJabatan(jabatanAlumni);
218
219 dataPekerjaan.setPendapatanPerbulan(gajiPerbulanAlumni);
220
221
222 DatabaseReference reference =
223 FirebaseDatabase.getInstance().getReference();
224 reference.child(Global.TRACER_STUDY)
225     .child(Global.UNIVERSITY)
226     .child(Global.FACULTY)
227     .child(Global.TYPE_DATA)
228     .child(Global.TYPE_USER)
229     .child(Global.TYPE_DATA_PKERJAAN)
230
231 .child(GlobalData.getSharedGlobalData().getUser().getNim())
232
233 .child(this.dataPekerjaan.getFirebaseId())
234 .setValue(dataPekerjaan);
235
236 }
237 return true;
238 }
239
240 @Override
241 public void onDateSet(DatePicker view, int year, int
242 month, int dayOfMonth) {
243     try {
244         Calendar calendar = Calendar.getInstance();
245         calendar.set(year, month, dayOfMonth);
246         SimpleDateFormat sdf = new
247 SimpleDateFormat(Global.DATE_DATE_STR_FORMAT);
248         String date = sdf.format(calendar.getTime());
249         if (requestCode == REQUEST_MULAI){
250             mulaiBekerja.setText(date);
251         } else if (requestCode == REQUEST_SELESAI){
252             selesaiBekerja.setText(date);
253         }
254     } catch (Exception e) {
255     }
256 }

```

Dalam tabel 6.1 menjelaskan tentang bagaimana proses implementasi dari memasukkan informasi pekerjaan, dimulai dari deklarasi kelas, deklarasi variabel dan deklarasi *method*. Deklarasi *method* `onCreate()` dilakukan sebagai awal dari jalannya program halaman masukkan informasi pekerjaan. Method `onCreate` akan membuat tampilan, lalu tampilan akan ditampilkan kepada alumni. Kemudian *method* `onGetViaCalendarMulaiClicked` akan dipanggil ketika pengguna menekan tombol yang ada dihalaman pertanyaan tambah informasi pekerjaan, yang mana akan memanggil tampilan `datetimepicker` yang nantinya alumni dapat mengubah tanggal mulai bekerja alumni. Begitu juga dengan *method* `onGetViaCalendarSelesaiClicked` akan dipanggil saat

pengguna menekan tombol buka kalender dari tampilan penanggalan dengan pertanyaan kapan alumni selesai bekerja. Kedua method tersebut akan mentrigger fungsi `onDateSet` yang mana akan melakukan proses validasi tanggal yang telah dipilih oleh alumni, lalu akan diubah menjadi format `dd/MM/yyyy` dan akan ditampilkan kepada alumni sebagai tanggal yang dipilih.

Ketika alumni menekan tombol *save record* maka akan mentrigger method `onSaveRecordClicked` dan `onSaveRecordClicked` akan memanggil method `validateCurrentPage` untuk memvalidasi masukan dari alumni, apakah sudah sesuai dengan format format yang diminta atau belum. Jika format yang dimasukkan oleh alumni tidak sesuai dengan yang dibutuhkan oleh sistem, maka sistem akan menampilkan pesan kesalahan yang dilakukan oleh alumni, serta akan menghentikan proses validasi dan proses penyimpanan ke *database*. Namun jika masukan alumni sesuai dengan format yang sudah ditentukan oleh sistem, maka sistem akan menampilkan toast data berhasil disimpan, terima kasih.

Method `onOptionsItemSelected` akan memproses jika alumni menekan tombol kembali yang ada dipojok kiri atas halaman, dan akan memanggil method `onBackPressed`, yang mentrigger fungsi `back` yang ada di bagian kanan bawah layar.

6.1.1.2 Implementasi Kode Program Masukkan Informasi Pendidikan

Tabel 6.2 Kode Masukkan Informasi Pendidikan

| No | Kode |
|----|---|
| 1 | <code>@Override</code> |
| 2 | <code>protected void onCreate(Bundle savedInstanceState) {</code> |
| 3 | <code>super.onCreate(savedInstanceState);</code> |
| 4 | |
| 5 | <code>setContentView(R.layout.activity_add_info_pendidikan);</code> |
| 6 | <code>ButterKnife.bind(this);</code> |
| 7 | |
| 8 | <code>final String[] listJurusan =</code> |
| 9 | <code>getResources().getStringArray(R.array.department_and_program</code> |
| 10 | <code>me);</code> |
| 11 | |
| 12 | <code>ArrayAdapter adapter =</code> |
| 13 | <code>ArrayAdapter.createFromResource(this,</code> |
| 14 | <code>R.array.department_and_programme, R.layout.item_spinner);</code> |
| 15 | |
| 16 | <code>adapter.setDropDownViewResource(R.layout.item_spinner_dropdo</code> |
| 17 | <code>wn);</code> |
| 18 | <code>dropdown.setAdapter(adapter);</code> |
| 19 | <code>jurusanTerpilih = listJurusan[0];</code> |
| 20 | <code>dropdown.setOnItemClickListener(new</code> |
| 21 | <code>AdapterView.OnItemClickListener() {</code> |
| 22 | <code>@Override</code> |
| 23 | <code>public void onItemClick(AdapterView<?></code> |
| 24 | <code>parent, View view, int position, long id) {</code> |
| 25 | <code>jurusanTerpilih = listJurusan[position];</code> |
| 26 | <code>}</code> |
| 27 | |
| 28 | <code>@Override</code> |

```

29         public void onNothingSelected(AdapterView<?>
30 parent) {
31
32         }
33     });
34     if (null !=
35 getIntent().getStringExtra(TOOLBAR_TITLE)) {
36         setSupportActionBar(toolbar);
37
38     getSupportActionBar().setTitle(getIntent().getStringExtra(TO
39 OLBAR_TITLE));
40     } else {
41         setSupportActionBar(toolbar);
42         getSupportActionBar().setTitle("Riwayat
43 Pendidikan");
44     }
45
46     getSupportActionBar().setDisplayHomeAsUpEnabled(true);
47 }
48
49 @Override
50 public boolean onOptionsItemSelected(MenuItem item) {
51     switch (item.getItemId()) {
52         case android.R.id.home:
53             onBackPressed();
54             return true;
55     }
56     return super.onOptionsItemSelected(item);
57 }
58
59 @OnClick(R.id.btn_info_pendidikan_save_record)
60 void onSaveRecordClicked(View view) {
61     if (validateCurrentPage()) {
62         Toast.makeText(this, "Data Berhasil
63 Disimpan\nTerima Kasih", Toast.LENGTH_SHORT).show();
64         finish();
65     }
66 }
67
68 private boolean validateCurrentPage() {
69     String universitasAlumni =
70 universitas.getText().toString();
71     String kotaAlumni = kota.getText().toString();
72     String negaraAlumni = negara.getText().toString();
73     String jurusanAlumni = jurusanTerpilih;
74     String tahunLulusAlumni =
75 tahunLulus.getText().toString();
76     RadioButton level =
77 findViewById(levelOfEducation.getCheckedRadioButtonId());
78     String education = null == level ? "" :
79 level.getText().toString();
80     String pendidikanAlumni =
81 degreeGraduates.getText().toString();
82
83     if (!Tools.isThisStringValid(universitasAlumni)) {
84         Toast.makeText(this,
85 getString(R.string.is_required,
86 getString(R.string.university)), Toast.LENGTH_SHORT).show();
87         return false;

```



```

88     }
89     if (!Tools.isThisStringValid(kotaAlumni)) {
90         Toast.makeText(this,
91 getString(R.string.is_required,    getString(R.string.city)),
92 Toast.LENGTH_SHORT).show();
93         return false;
94     }
95     if (!Tools.isThisStringValid(negaraAlumni)) {
96         Toast.makeText(this,
97 getString(R.string.is_required,
98 getString(R.string.country)), Toast.LENGTH_SHORT).show();
99         return false;
100    }
101    if (!Tools.isThisStringValid(jurusanAlumni)) {
102        Toast.makeText(this,
103 getString(R.string.is_required,
104 getString(R.string.department_and_programme)),
105 Toast.LENGTH_SHORT).show();
106        return false;
107    }
108    if (!Tools.isThisStringValid(tahunLulusAlumni)) {
109        Toast.makeText(this,
110 getString(R.string.is_required,
111 getString(R.string.graduated_years)),
112 Toast.LENGTH_SHORT).show();
113        return false;
114    }
115    if (!Tools.isThisStringValid(education)) {
116        Toast.makeText(this,
117 getString(R.string.is_required,
118 getString(R.string.level_of_educational)),
119 Toast.LENGTH_SHORT).show();
120        return false;
121    }
122    if (!Tools.isThisStringValid(pendidikanAlumni)) {
123        Toast.makeText(this,
124 getString(R.string.is_required,
125 getString(R.string.degree_graduates)),
126 Toast.LENGTH_SHORT).show();
127        return false;
128    }
129
130
131    if (null == this.dataPendidikan){
132        DataPendidikan      dataPendidikan      =      new
133 DataPendidikan();
134        dataPendidikan.setId(Tools.getCurrentTime());
135
136        dataPendidikan.setNim(GlobalData.getSharedGlobalData().getUs
137 er().getNim());
138
139        dataPendidikan.setUniversitas(universitasAlumni);
140        dataPendidikan.setKota(kotaAlumni);
141        dataPendidikan.setNegara(negaraAlumni);
142        dataPendidikan.setProgramStudi(jurusanAlumni);
143        dataPendidikan.setTahun_lulus(tahunLulusAlumni);
144        dataPendidikan.setLevelPendidikan(education);
145
146        dataPendidikan.setGelar_diperoleh(pendidikanAlumni);

```

```

147
148         reference =
149         FirebaseDatabase.getInstance().getReference();
150         reference.child(Global.TRACER_STUDY)
151             .child(Global.UNIVERSITY)
152             .child(Global.FACULTY)
153             .child(Global.TYPE_DATA)
154             .child(Global.TYPE_USER)
155             .child(Global.TYPE_DATA_PENDIDIKAN)
156
157         .child(GlobalData.getSharedGlobalData().getUser().getNim())
158             .push()
159             .setValue(dataPendidikan);
160     } else {
161         DataPendidikan dataPendidikan = new
162         DataPendidikan();
163
164         dataPendidikan.setId(this.dataPendidikan.getId());
165
166         dataPendidikan.setNim(this.dataPendidikan.getNim());
167
168         dataPendidikan.setUniversitas(universitasAlumni);
169         dataPendidikan.setKota(kotaAlumni);
170         dataPendidikan.setNegara(negaraAlumni);
171         dataPendidikan.setProgramStudi(jurusanAlumni);
172         dataPendidikan.setTahun_lulus(tahunLulusAlumni);
173         dataPendidikan.setLevelPendidikan(education);
174
175         dataPendidikan.setGelar_diperoleh(pendidikanAlumni);
176
177         reference =
178         FirebaseDatabase.getInstance().getReference();
179         reference.child(Global.TRACER_STUDY)
180             .child(Global.UNIVERSITY)
181             .child(Global.FACULTY)
182             .child(Global.TYPE_DATA)
183             .child(Global.TYPE_USER)
184             .child(Global.TYPE_DATA_PENDIDIKAN)
185
186         .child(GlobalData.getSharedGlobalData().getUser().getNim())
187
188         .child(this.dataPendidikan.getFirebaseId())
189             .setValue(dataPendidikan);
190     }
191
192     return true;
193 }
194
195 @Override
196 protected void onDestroy() {
197     super.onDestroy();
198     reference = null;
199 }
200 }

```

Dalam tabel 6.2 menjelaskan tentang bagaimana proses implementasi dari memasukkan informasi pendidikan, dimulai dari deklarasi kelas, deklarasi variabel dan deklarasi *method*. Deklarasi *method* `onCreate()` dilakukan sebagai awal dari jalannya program halaman masukkan informasi pendidikan. *Method*

`onCreate` akan membuat tampilan serta melakukan beberapa hal seperti proses membuat list jurusan, memasang judul halaman yang ada di atas tampilan, lalu tampilan itu akan ditampilkan kepada alumni.

Method `onOptionsItemSelected` akan memproses jika alumni menekan tombol kembali yang ada dipojok kiri atas halaman, dan akan memanggil method `onBackPressed`, yang mentrigger fungsi back yang ada di bagian kanan bawah layar.

Ketika alumni menekan tombol *save record* maka akan mentrigger *method* `onSaveRecordClicked` dan `onSaveRecordClicked` akan memanggil *method* `validateCurrentPage` untuk memvalidasi masukan dari alumni, apakah sudah sesuai dengan format format yang diminta atau belum. Jika format yang dimasukkan oleh alumni tidak sesuai dengan yang dibutuhkan oleh sistem, maka sistem akan menampilkan pesan kesalahan yang dilakukan oleh alumni, serta akan menghentikan proses validasi dan proses penyimpanan ke *database*. Namun jika masukan alumni sesuai dengan format yang sudah ditentukan oleh sistem, maka sistem akan menampilkan toast data berhasil disimpan, terima kasih.

Method `onDestroy()` dilakukan hapus memori dengan membuat nilai reference menjadi null sehingga tidak banyak memakan baterai dan tidak banyak memakan memori.

Tabel 6.3 Kode Melihat Data Pekerjaan

| No | Kode |
|----|--|
| 1 | |
| 2 | <code>@Override</code> |
| 3 | <code>protected void onCreate(Bundle savedInstanceState) {</code> |
| 4 | <code>super.onCreate(savedInstanceState);</code> |
| 5 | |
| 6 | <code>setContentView(R.layout.activity_lihat_info_pekerjaan);</code> |
| 7 | <code>ButterKnife.bind(this);</code> |
| 8 | <code>dataPekerjaan = (DataPekerjaan)</code> |
| 9 | <code>getIntent().getSerializableExtra(ID_PERTANYAAN);</code> |
| 10 | <code>checkDataForCurrentPage(dataPekerjaan);</code> |
| 11 | <code>setSupportActionBar(toolbar);</code> |
| 12 | |
| 13 | <code>getSupportActionBar().setDisplayHomeAsUpEnabled(true);</code> |
| 14 | <code>}</code> |
| 15 | |
| 16 | <code>@Override</code> |
| 17 | <code>public boolean onOptionsItemSelected(MenuItem item) {</code> |
| 18 | <code>switch (item.getItemId()) {</code> |
| 19 | <code>case android.R.id.home:</code> |
| 20 | <code>onBackPressed();</code> |
| 21 | <code>return true;</code> |
| 22 | <code>}</code> |
| 23 | <code>return super.onOptionsItemSelected(item);</code> |
| 24 | <code>}</code> |
| 25 | |
| 26 | <code>private void checkDataForCurrentPage(DataPekerjaan</code> |
| 27 | <code>dataPekerjaan) {</code> |

```

28 tanggalMulai.setText(dataPekerjaan.getMulaiBekerja());
29
30 tanggalSelesai.setText(dataPekerjaan.getSelesaiBekerja());
31
32 statusPekerjaan.setText(dataPekerjaan.getStatusPekerjaan());
33
34 bidangPekerjaan.setText(dataPekerjaan.getBidangPekerjaan());
35
36 sesuaiKeahlian.setText(dataPekerjaan.getSesuaiKeahlian());
37
38 namaPerusahaan.setText(dataPekerjaan.getNamaPerusahaan());
39 jabatan.setText(dataPekerjaan.getJabatan());
40
41 pendapatan.setText(dataPekerjaan.getPendapatanPerbulan());
42
43 masihBekerja.setText(dataPekerjaan.getMasihBekerja().equals(
44 "1") ? getString(R.string.s_yes) :
45 getString(R.string.s_no));
46
47 pekerjaanPertama.setText(dataPekerjaan.getPekerjaanPertama().equals("1") ?
48 getString(R.string.s_yes) :
49 getString(R.string.s_no));
50
51 }
52
53 @OnClick (R.id.btn_lihat_info_pekerjaan_edit_record)
54 void editRecordOnClickListener(View view){
55     Intent intent = new Intent(getApplicationContext(),
56 AddInfoPekerjaanActivity.class);
57     intent.putExtra(TOOLBAR_TITLE, "Riwayat Pekerjaan");
58     intent.putExtra(ID_PERTANYAAN, dataPekerjaan);
59     startActivity(intent);
60 }

```

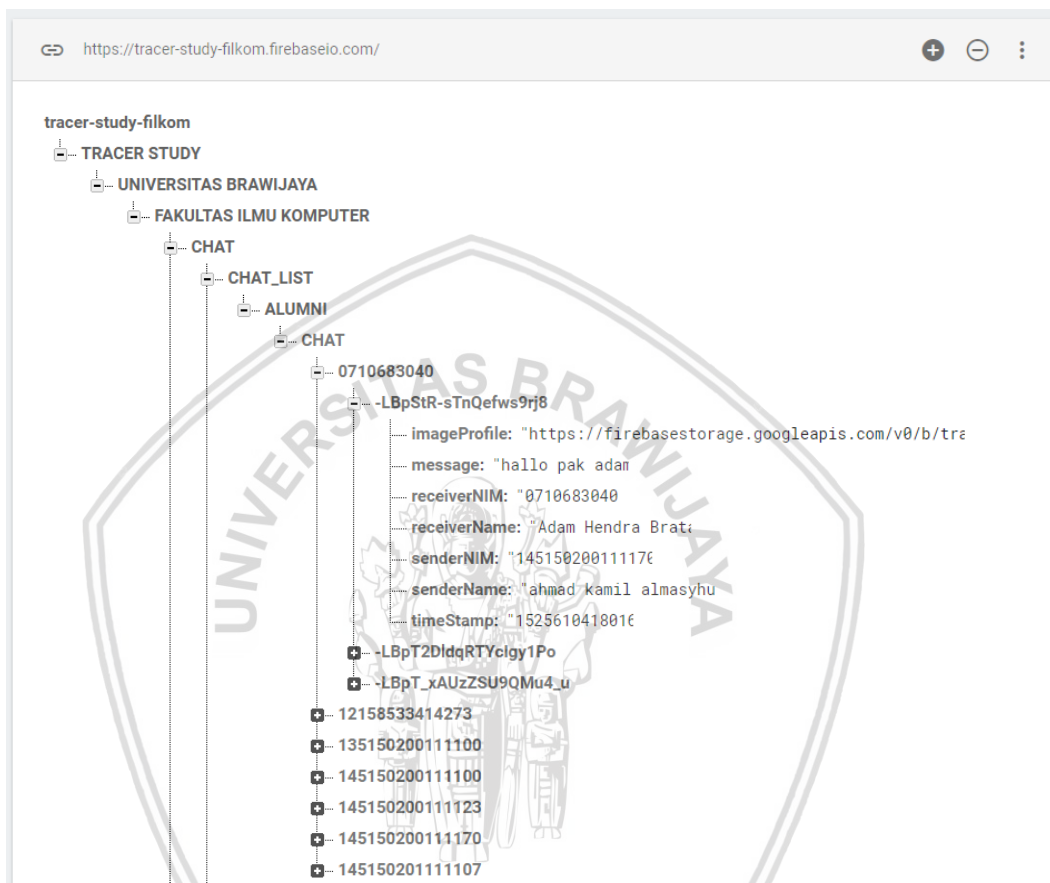
Dalam tabel 6.3 menjelaskan tentang bagaimana menampilkan data pekerjaan yang telah diisi oleh alumni dari list data pekerjaan yang ada, dimulai dari deklarasi kelas, deklarasi variabel dan deklarasi *method*. Deklarasi *method* `onCreate()` dilakukan sebagai awal dari jalannya program halaman masukkan informasi pendidikan. *Method* `onCreate` akan membuat tampilan serta melakukan beberapa hal seperti proses pengambilan data yang dikirim dari halaman sebelumnya menggunakan `bundle serialize`, serta memasang judul halaman yang ada di atas tampilan, lalu tampilan itu akan ditampilkan kepada alumni.

Method `onOptionsItemSelected` akan memproses jika alumni menekan tombol kembali yang ada dipojok kiri atas halaman, dan akan memanggil *method* `onBackPressed`, yang mentrigger fungsi `back` yang ada di bagian kanan bawah layar.

Method `checkDataForCurrentPage` akan melakukan proses input secara otomatis oleh program kedalam `field` yang sudah ada dalam halaman lihat riwayat pekerjaan. Sedangkan *method* `editRecordOnClickListener` akan melakukan proses perpindahan halaman ke halaman edit riwayat pekerjaan dimana akan menampilkan data-data yang sudah ada di halaman lihat data pekerjaan.

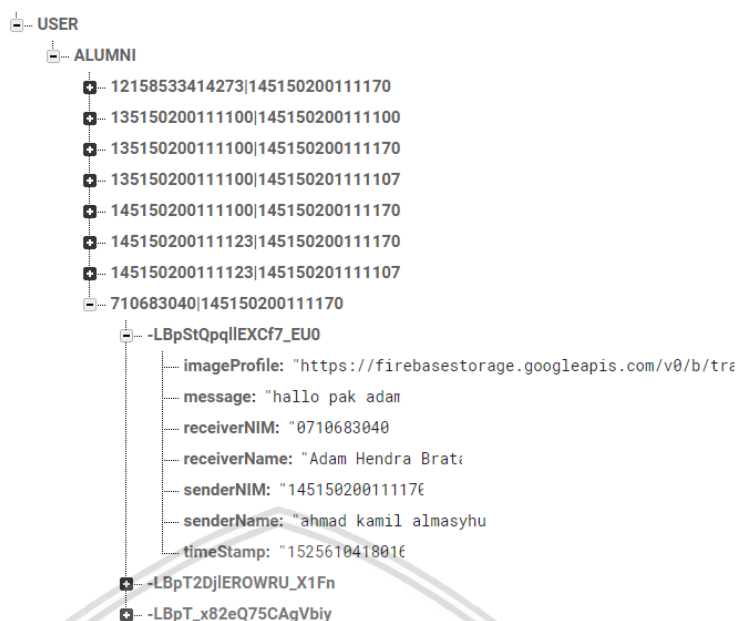
6.1.2 Implementasi Data

Berikut ini adalah bentuk data hasil dari implementasi yang telah dilakukan menggunakan firebase *realtime database* sehingga dapat menjadi sebuah sistem tracer-study yang utuh dan dapat melakukan beberapa hal utama seperti *chat*, membuka lowongan pekerjaan, menyimpan data diri, menyimpan informasi pekerjaan, menyimpan informasi pendidikan dari alumni.



Gambar 6.1 Implementasi Data *List Chat Alumni*

Dalam gambar 6.1 menjelaskan tentang bagaimana alumni dapat saling bertukar pesan dan saling mengetahui informasi terbaru tentang *chat* alumni lainnya kepada dirinya. Implementasi data ini digunakan dengan fitur *firebase push notification* yang dapat memberikan langsung pesan penerima kepada penerima sehingga penerima tidak perlu untuk mengecek data di firebase *realtime database* setiap waktu sehingga akan menjadi lebih hemat baterai dan lebih hemat penggunaan memori karena tidak adanya *background* proses yang memakan banyak *tread* dan memakan banyak memori serta baterai.



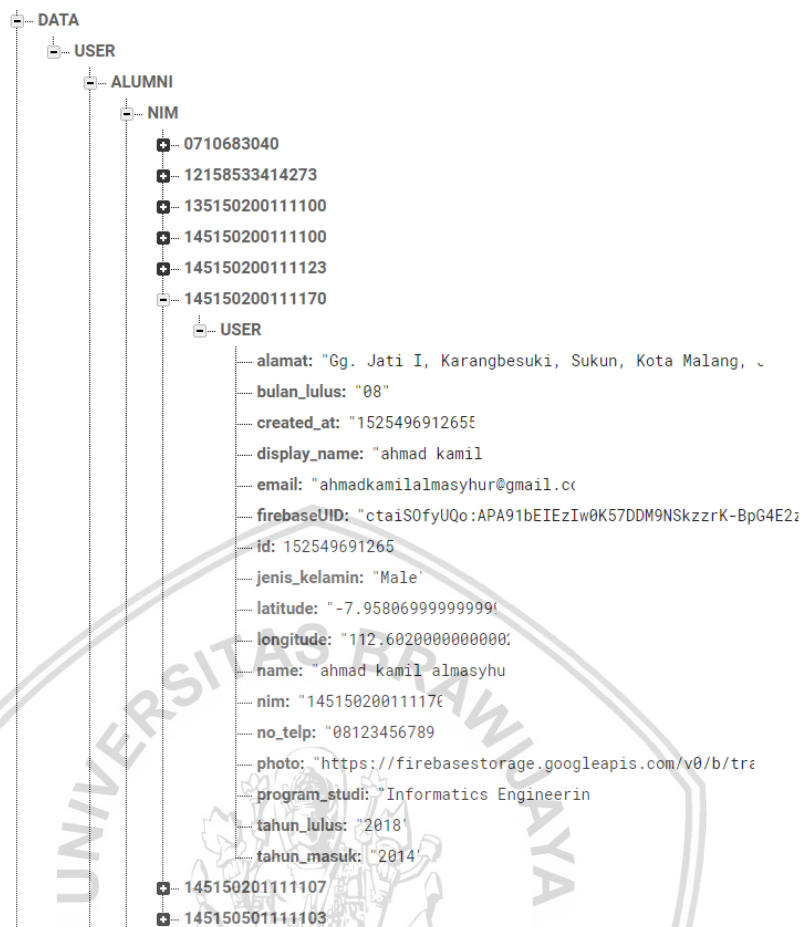
Gambar 6.2 Implementasi Data *Chat* Alumni

Dalam gambar 6.2 digunakan untuk proses pertukaran pesan tanpa menggunakan *firebase push notification* sehingga alumni bisa saling berukar pesan dan mengetahui perubahan dengan mendengarkan perubahan yang ada di salah satu node anak. Dengan menggunakan metode ini, alumni dapat saling terhubung dengan alumni lainnya sesuai dengan *key node* yang ada.

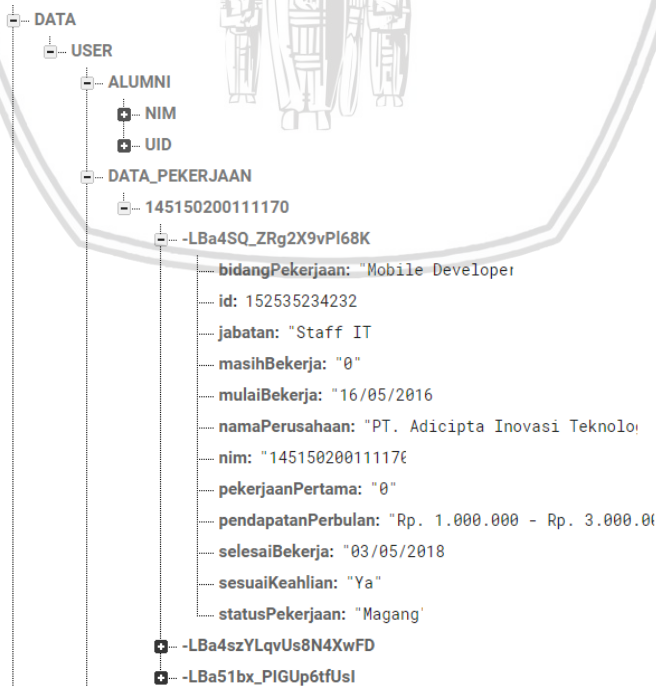
Dalam gambar 6.3 menjelaskan tentang data user alumni berdasarkan nim yang dimiliki oleh alumni, data yang dimiliki oleh alumni berupa alamat, bulan_lulus, *created_at*, *display_name*, *email*, *firebaseUID*, id, jenis_kelamin, *latitude*, *longitude*, *name*, nim, no_telp, *photo*, *program_studi*, tahun_lulus, tahun_masuk.

Dalam gambar 6.4 menjelaskan tentang data pekerjaan yang dimiliki oleh alumni berdasarkan nim yang dimiliki alumni, dimana data yang dimiliki oleh alumni dapat lebih dari 1 data untuk data pekerjaan.

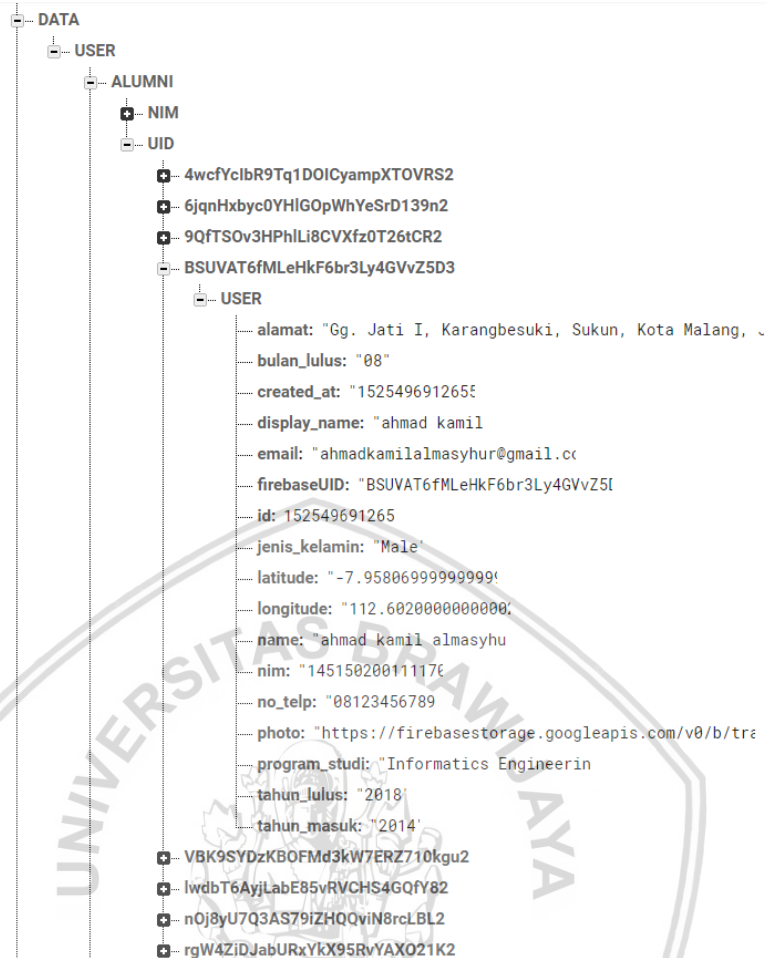
Dalam gambar 6.5 menjelaskan tentang data alumni yang disimpan berdasarkan *UID* alumni sehingga dapat diakses dengan nim ataupun dengan *UID* alumni.



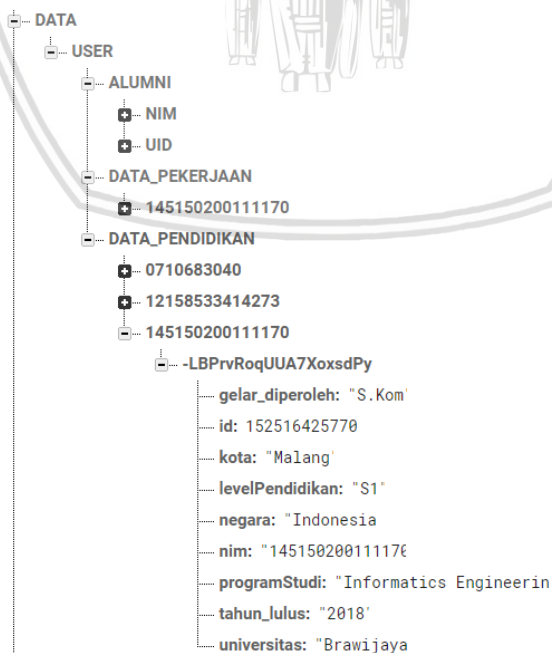
Gambar 6.3 Implementasi Data User Alumni



Gambar 6.4 Implementasi Data Pekerjaan Alumni



Gambar 6.5 Implementasi Data User Alumni



Gambar 6.6 Implementasi Data Pendidikan Alumni



Gambar 6.7 Implementasi Data Timeline untuk Alumni

Dalam gambar 6.6 menjelaskan tentang data yang dimiliki oleh alumni berdasarkan nim yang dimiliki oleh alumni, dimana data dapat lebih dari satu data pendidikan.

Dalam gambar 6.7 menjelaskan tentang data lowongan pekerjaan yang akan ditampilkan kepada alumni, mulai dari alamat perusahaan, deskripsi pekerjaan, detail deskripsi pekerjaan, jenis pekerjaan, kontak yang dapat dihubungi, logo perusahaan, nama perusahaan, status pekerjaan, syarat untuk mendaftar, syarat untuk bekerja, syarat pengalaman kerja, serta batas waktu pendaftaran untuk lowongan pekerjaan tersebut. Alumni juga dapat melihat lokasi perusahaan dengan menggunakan latitude dan longitude dari perusahaan tersebut.

Setelah implementasi data dijelaskan maka akan dilanjutkan dengan implementasi antarmuka yang mana akan ditampilkan tampilan antarmuka yang sudah jadi dalam bentuk *screenshot* dari sistem *tracer study*. Antarmuka yang dilampirkan merupakan antarmuka utama, beberapa antarmuka sistem *tracer study* tidak dilampirkan karena dirasa sudah diwakilkan oleh beberapa antarmuka berikut ini :

6.1.3 Implementasi Antarmuka

6.1.3.1 Implementasi Antarmuka Halaman *Chat*



Gambar 6.8 Antarmuka Halaman *Chat*

Dalam gambar 6.8 digambarkan nama pengguna sistem yang lain, yang sudah pernah melakukan *chat* kepada pengguna yang sedang login, serta ditampilkan pula, apa pesan terakhir teman alumni, dan kapan pesan terakhir yang dikirimkan oleh teman alumni, dalam halaman ini, alumni dapat melihat beberapa tab tambahan yaitu tab LoKer atau lowongan pekerjaan, tab profil serta tanda +, dimana tanda + digunakan untuk mencari teman alumni yang lain untuk saling bertukar pesan. Halaman LoKer akan dijelaskan lebih lanjut pada antarmuka halaman LoKer, dan antarmuka halaman profil akan dijelaskan lebih lanjut pada antarmuka halaman profil

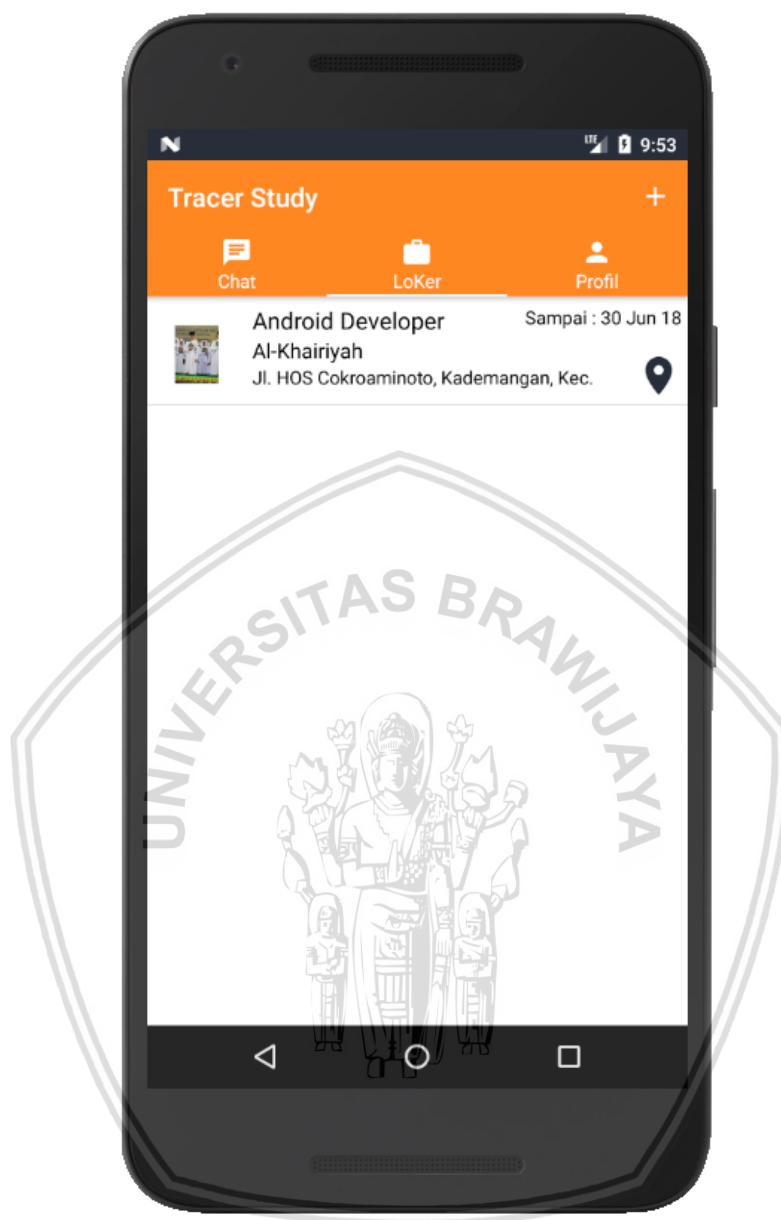
6.1.3.2 Implementasi Antarmuka Halaman Ruang *Chat*



Gambar 6.9 Antarmuka Halaman Ruang *Chat*

Dalam gambar 6.9 digambarkan nama teman alumni di atas halaman, tombol kembali untuk kemudahan berpindah halaman bagi alumni, *field* masukkan pesan untuk menulis pesan, tombol kirim ada disebelah kanan, dan tombol lampiran *file* ada disebelah kiri halaman.

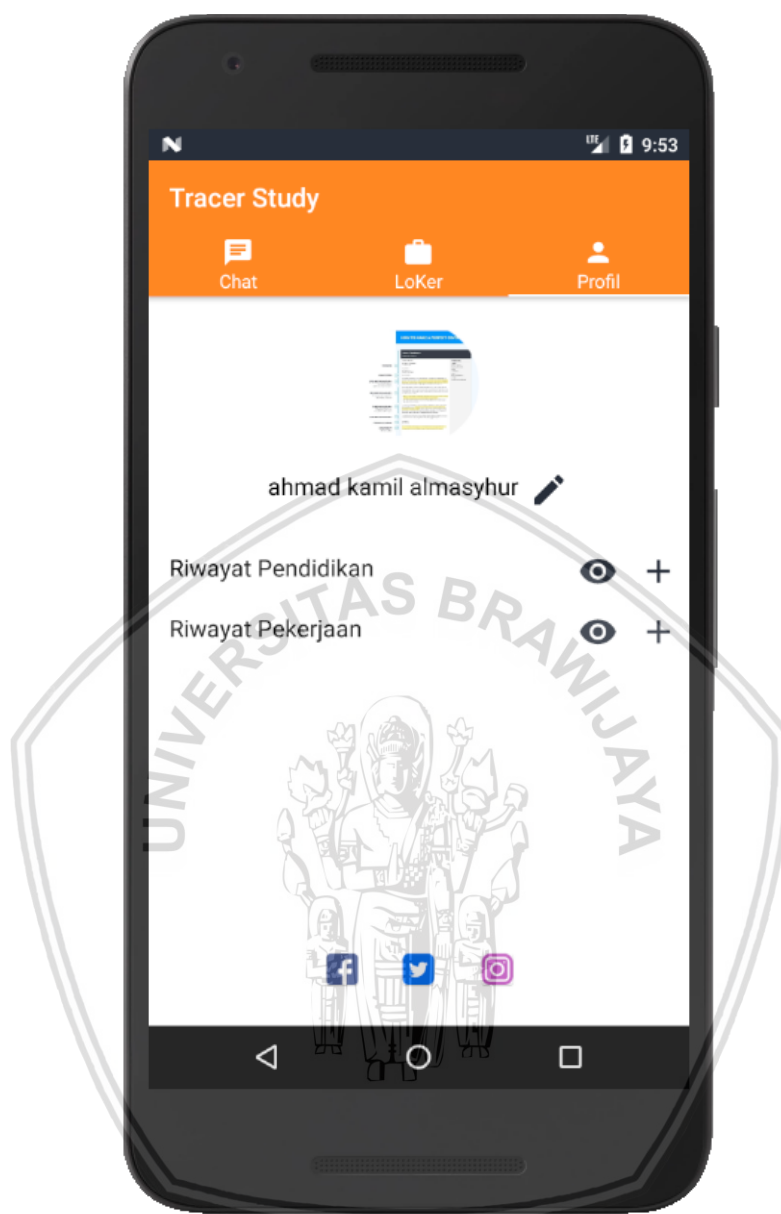
6.1.3.3 Implementasi Antarmuka Halaman Lowongan Pekerjaan



Gambar 6.10 Antarmuka Halaman Lowongan Pekerjaan

Dalam gambar 6.10 digambarkan tampilan tab seperti yang ada dalam gambar 6.8. Pada halaman lowongan pekerjaan, dituliskan jenis pekerjaan yang sedang dibuka oleh perusahaan, nama perusahaan yang sedang membuka lowongan pekerjaan terdapat dibawah jenis pekerjaan, lalu diikuti oleh alamat perusahaan, serta tombol lokasi jika alumni ingin mengetahui lokasi perusahaan dengan lebih presisi. Terdapat tulisan sampai 30 Jun 18 merupakan batas akhir waktu pendaftaran bagi alumni untuk mendaftar pada lowongan pekerjaan yang dimaksud.

6.1.3.4 Implementasi Antarmuka Halaman Lowongan Pekerjaan



Gambar 6.11 Antarmuka Halaman Profil

Dalam gambar 6.11 digambarkan tampilan tab seperti yang ada Digambar 6.8, dimana pada halaman ini, alumni dapat mengubah data diri alumni, menambahkan riwayat pendidikan alumni, menambahkan riwayat pekerjaan alumni, serta melihat riwayat pendidikan serta riwayat pekerjaan yang telah ditambahkan oleh alumni pada sesi-sesi sebelumnya.

6.1.3.5 Implementasi Antarmuka Halaman Riwayat Pendidikan

The screenshot shows a mobile application interface for adding education history. The title bar is orange and labeled 'Pendidikan'. The form contains the following fields:

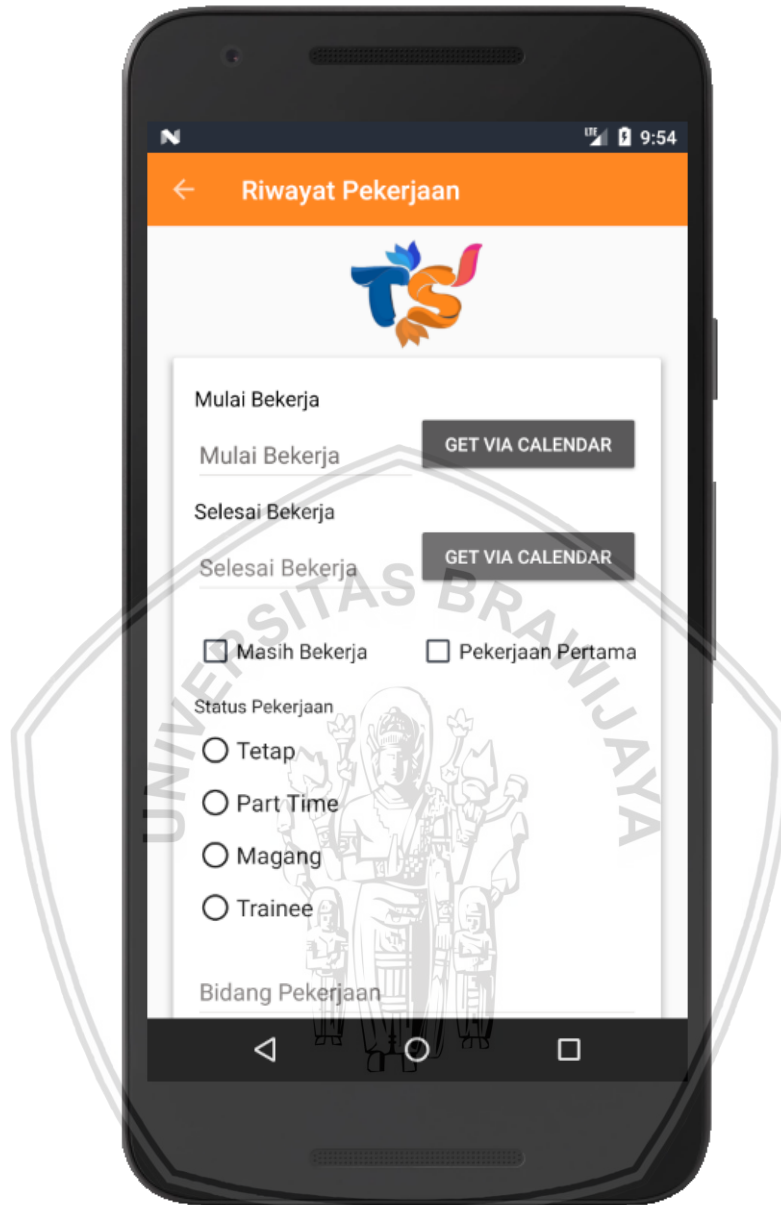
- Universitas: A text input field.
- Kota: A text input field.
- Negara: A text input field.
- Department and Programme: A dropdown menu currently showing 'Teknik Informatika'.
- Tahun Lulus: A text input field.
- Jenjang Pendidikan: Three radio button options labeled S1, S2, and S3.
- Gelar yang diperoleh: A text input field.

At the bottom of the form is an orange button labeled 'SIMPAN RIWAYAT'. The entire interface is displayed on a black smartphone frame. A large, semi-transparent watermark of the Universitas Brawijaya logo is centered over the form.

Gambar 6.12 Antarmuka Halaman Tambah Riwayat Pendidikan

Dalam gambar 6.12 alumni dapat mengisi beberapa pertanyaan yang sudah disediakan oleh sistem *tracer study*, yang mana data yang telah diisikan oleh alumni dapat dilihat dan diedit oleh alumni.

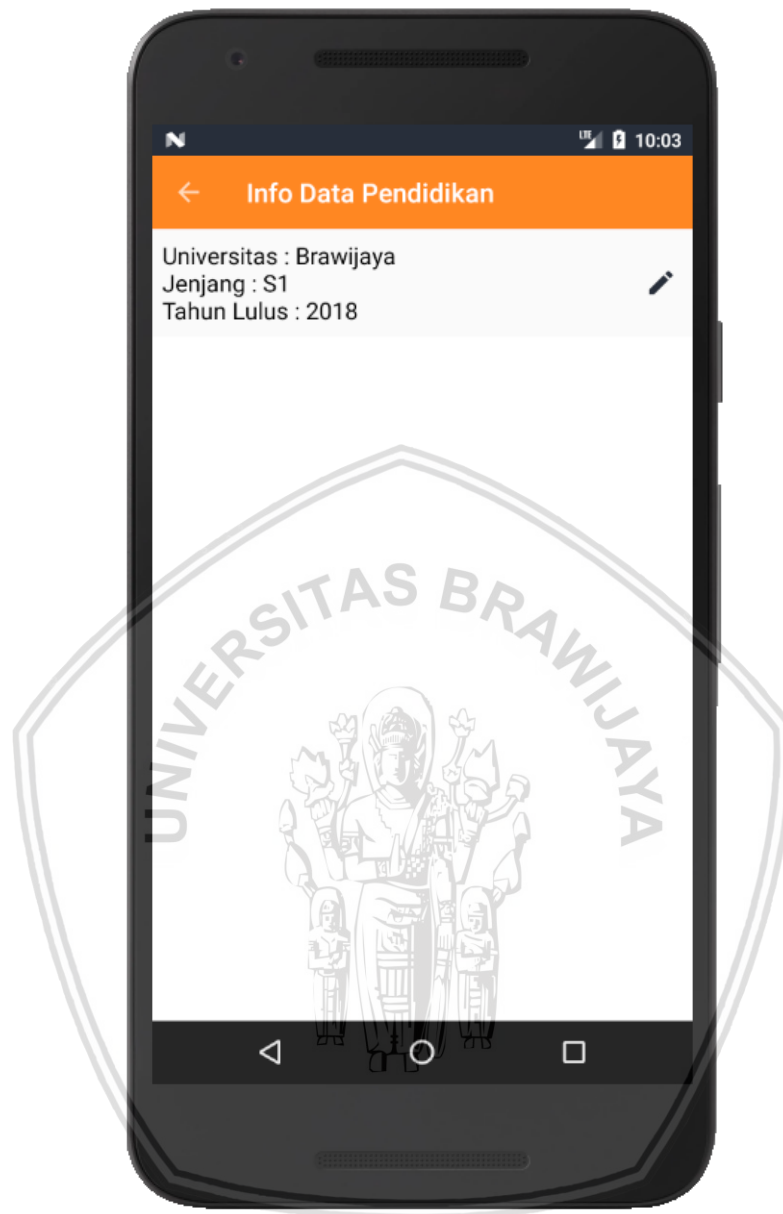
6.1.3.6 Implementasi Antarmuka Halaman Riwayat Pendidikan



Gambar 6.13 Antarmuka Halaman Tambah Riwayat Pekerjaan

Dalam gambar 6.13 alumni dapat mengisi beberapa pertanyaan yang sudah disediakan oleh sistem *tracer study*, yang mana data yang telah diisikan oleh alumni dapat dilihat dan diedit oleh alumni.

6.1.3.7 Implementasi Antarmuka Halaman Riwayat Pendidikan



Gambar 6.14 Antarmuka Halaman Lihat Riwayat Pendidikan

Dalam gambar 6.14 alumni dapat melihat serta mengedit data yang telah alumni isikan sebelumnya, sesuai dengan data yang dapat kita lihat di gambar 6.12.

6.1.3.8 Implementasi Antarmuka Halaman Riwayat Pendidikan



Gambar 6.15 Antarmuka Halaman Lihat Riwayat Pekerjaan

Dalam gambar 6.15 alumni dapat melihat serta mengedit data yang telah alumni isikan sebelumnya, sesuai dengan data yang dapat kita lihat di gambar 6.13.

BAB 7 PENGUJIAN

Setelah tahap implementasi selesai dilakukan, maka akan dilakukan pengujian yang bertujuan untuk mengetahui apakah aplikasi yang telah dibangun telah sesuai dengan kebutuhan perangkat lunak yang telah didapatkan pada fase analisis kebutuhan, pengujian yang digunakan adalah pengujian fungsional, kompatibilitas dan pengujian usabilitas.

7.1 Pengujian

Setiap pengujian akan dilakukan bertahap dimana hasil pengujian akan diletakkan pada sub bab 7.2 pada bab ini, sedangkan untuk tahapan pengujian akan diletakkan pada sub bab 7.1, serta menampilkan tabel yang dibutuhkan untuk melakukan proses pengujian.

Pengujian ini menggunakan tiga jenis perangkat keras yang berbeda dan masing masing perangkat keras memiliki spesifikasi perangkat lunak yang berbeda serta menggunakan satu emulator Android yang telah disediakan oleh aplikasi pengembangan Android Studio.

Tabel 7.1 Spesifikasi Perangkat Keras Android Samsung Galaxy S6

| Perangkat Keras Android | |
|-------------------------|---|
| Chipset | Exynos 7420 Octa |
| CPU | Octa-core (4x2.1 GHz Cortex-A57 & 4x1.5 GHz Cortex-A53) |
| Kapasitas RAM | 3 GB RAM |
| Kapasitas Memori | 32 GB |
| GPU | Mali-T760MP8 |
| Resolusi Layar | 1440 x 2560 pixels, 16:9 Rasio |
| Densitas resolusi | ~577 ppi |

Dalam tabel 7.1 menjelaskan bagian dari spesifikasi hp yang mempengaruhi kualitas sistem *tracer study* yang akan digunakan / di uji nantinya. Perbedaan kapasitas *ram*, perbedaan memori, perbedaan *cpu* dan *chipset* akan mempengaruhi bagaimana sistem bertindak kepada perangkat keras. Sebuah Aplikasi harus dapat digunakan di lingkungan yang berbeda (seperti berbeda perangkat keras, berbeda koneksi, berbeda perangkat lunak (sistem operasi), berbeda *browser*) dan harus didesain untuk kompatibel dengan semua (Pressman, 2008).

Tabel 7.2 Spesifikasi Perangkat Keras Android Samsung J2 Pro (2016)

| Perangkat Keras Android | |
|-------------------------|------------------------------|
| Chipset | Spreadtrum SC8830 |
| CPU | Quad-core 1.5 GHz Cortex-A7 |
| Kapasitas RAM | 2 GB RAM |
| Kapasitas Memori | 16 GB |
| GPU | Mali-400MP2 |
| Resolusi Layar | 720 x 1280 pixels 16:9 Rasio |
| Densitas resolusi | ~294 ppi |

Dalam tabel 7.2 menjelaskan bagian dari spesifikasi hp yang mempengaruhi kualitas sistem *tracer study* yang akan digunakan / di uji nantinya. Perbedaan kapasitas *ram*, perbedaan memori, perbedaan *cpu* dan *chipset* akan mempengaruhi bagaimana sistem bertindak kepada perangkat keras. Sebuah Aplikasi harus dapat digunakan di lingkungan yang berbeda (seperti berbeda perangkat keras, berbeda koneksi, berbeda perangkat lunak (sistem operasi), berbeda *browser*) dan harus didesain untuk kompatibel dengan semua (Pressman, 2008).

Tabel 7.3 Spesifikasi Perangkat Keras Android Emulator

| Perangkat Keras Android Emulator | |
|----------------------------------|-------------------------------|
| Chipset | - |
| CPU | - |
| Kapasitas RAM | 1536 MB RAM |
| Kapasitas Memori | ~1 GB |
| GPU | Auto |
| Resolusi Layar | 1920 x 1080 pixels 16:9 Rasio |
| Densitas resolusi | ~420 ppi |

Dalam tabel 7.3 menjelaskan bagian dari spesifikasi emulator yang mempengaruhi kualitas sistem *tracer study* yang akan digunakan / di uji nantinya. Perbedaan kapasitas *ram*, perbedaan memori, perbedaan *cpu* dan *chipset* akan mempengaruhi bagaimana sistem bertindak kepada perangkat keras. Sebuah Aplikasi harus dapat digunakan di lingkungan yang berbeda (seperti berbeda perangkat keras, berbeda koneksi, berbeda perangkat lunak (sistem operasi), berbeda *browser*) dan harus didesain untuk kompatibel dengan semua (Pressman, 2008).

Spesifikasi perangkat lunak yang ada disetiap perangkat keras diatas akan dijelaskan lebih lengkap dalam tabel 7.4 berikut ini:

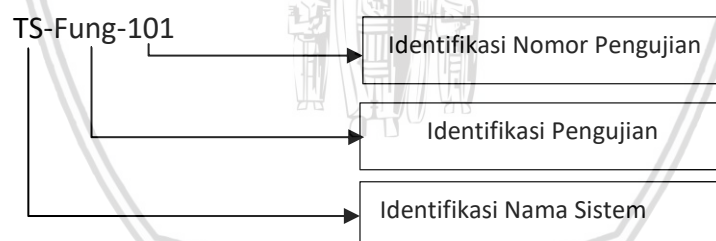
Tabel 7.4 Tabel Sistem Operasi Perangkat Keras

| Tipe Perangkat Keras | Sistem Operasi Perangkat Keras |
|------------------------------|--|
| Samsung Galaxy S6 | Android 7.0 (<i>Nougat</i>) |
| Samsung Galaxy J2 Pro (2016) | Android 6.0.1 (<i>Marshmallow</i>) |
| Android Emulator | Android 7.0 (<i>Nougat</i> dengan Google API) |

7.1.1 Pengujian Fungsional

Pengujian fungsional adalah pengujian yang dilakukan untuk memastikan apakah kebutuhan fungsional yang ada pada kebutuhan dapat divalidasi dan dipastikan dapat berjalan sesuai dengan kebutuhan yang telah ditentukan sebelumnya oleh pengembang aplikasi tracer sudy. Aktivitas ini menggunakan metode pengujian *blackbox testing* dengan mengacu pada skenario *use case* yang telah dipaparkan sebelumnya.

Pengujian fungsional akan dilakukan dengan cara mendeskripsikan kasus uji terlebih dahulu kemudian dilakukan proses pengujian, lalu akan dibandingkan kasus uji dengan hasil yang didapatkan pada pengujian. Lalu akan dibuat persentase yang akan membuktikan apakah sistem layak untuk digunakan atau tidak. Setiap kasus uji akan memiliki kode khusus dan unik yang memiliki aturan sebagai berikut:

**Gambar 7.1 Aturan Penomoran Pengujian Fungsional****Tabel 7.5 Kasus Uji Login**

| | |
|------------------|--|
| Nomor Kasus Uji | TS-Fung-111 |
| Nama Kasus Uji | Kasus Uji <i>Login</i> |
| Kode Kebutuhan | TS-F-G-101 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan sistem dapat menampilkan halaman awal / halaman utama pada aplikasi (halaman <i>chat</i>) setelah melakukan <i>login</i> guest dengan memasukkan <i>email</i> dan <i>password</i> dengan benar |

| | |
|-----------------------|--|
| Kasus Uji | Pengujian dilakukan dengan memasukkan <i>email</i> yang tidak sesuai |
| Prosedur pengujian | 1. Penguji membuka aplikasi <i>tracer study</i> dan masuk kehalaman <i>login</i> |
| | 2. Penguji memasukkan <i>email</i> tanpa kode @ |
| Hasil yang diharapkan | Sistem menampilkan informasi <i>login</i> gagal, <i>email</i> tidak valid |

Tabel 7.6 Kasus Uji Login

| | |
|-----------------------|--|
| Nomor Kasus Uji | TS-Fung-112 |
| Nama Kasus Uji | Kasus Uji Login |
| Kode Kebutuhan | TS-F-G-101 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan sistem dapat menampilkan halaman awal / halaman utama pada aplikasi (halaman <i>chat</i>) setelah melakukan <i>login guest</i> dengan memasukkan <i>email</i> dan <i>password</i> dengan benar |
| Kasus Uji | Pengujian dilakukan dengan memasukkan <i>password</i> yang tidak sesuai |
| Prosedur pengujian | 1. Penguji membuka aplikasi <i>tracer study</i> dan masuk kehalaman <i>login</i> |
| | 2. Penguji memasukkan <i>email</i> yang sesuai dan <i>password</i> kurang dari 6 digit |
| Hasil yang diharapkan | Sistem menampilkan informasi <i>login</i> gagal, <i>password</i> tidak valid |

Tabel 7.7 Kasus Uji Login

| | |
|------------------|--|
| Nomor Kasus Uji | TS-Fung-113 |
| Nama Kasus Uji | Kasus Uji Login |
| Kode Kebutuhan | TS-F-G-101 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan sistem dapat menampilkan halaman awal / halaman utama pada aplikasi (halaman <i>chat</i>) setelah melakukan <i>login guest</i> dengan memasukkan <i>email</i> dan <i>password</i> dengan benar |
| Kasus Uji | Pengujian dilakukan dengan memasukkan <i>password</i> |

| | |
|-----------------------|--|
| | yang sesuai tanpa mengisi <i>email</i> |
| Prosedur pengujian | 1. Penguji membuka aplikasi <i>tracer study</i> dan masuk kehalaman <i>login</i> |
| | 2. Penguji memasukkan <i>password</i> tanpa <i>email</i> |
| Hasil yang diharapkan | Sistem menampilkan informasi <i>login</i> gagal, <i>email</i> harus diisi |

Tabel 7.8 Kasus Uji Login

| | |
|-----------------------|--|
| Nomor Kasus Uji | TS-Fung-114 |
| Nama Kasus Uji | Kasus Uji Login |
| Kode Kebutuhan | TS-F-G-101 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan sistem dapat menampilkan halaman awal / halaman utama pada aplikasi (halaman <i>chat</i>) setelah melakukan <i>login guest</i> dengan memasukkan <i>email</i> dan <i>password</i> dengan benar |
| Kasus Uji | Pengujian dilakukan dengan memasukkan <i>email</i> yang sesuai dan <i>password</i> tidak diisi |
| Prosedur pengujian | 1. Penguji membuka aplikasi <i>tracer study</i> dan masuk kehalaman <i>login</i> |
| | 2. Penguji memasukkan <i>email</i> tanpa <i>password</i> |
| Hasil yang diharapkan | Sistem menampilkan informasi <i>login</i> gagal, <i>password</i> harus diisi |

Tabel 7.9 Kasus Uji Login

| | |
|--------------------|--|
| Nomor Kasus Uji | TS-Fung-115 |
| Nama Kasus Uji | Kasus Uji Login |
| Kode Kebutuhan | TS-F-G-101 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan sistem dapat menampilkan halaman awal / halaman utama pada aplikasi (halaman <i>chat</i>) setelah melakukan <i>login guest</i> dengan memasukkan <i>email</i> dan <i>password</i> dengan benar |
| Kasus Uji | Pengujian dilakukan dengan memasukkan <i>email</i> yang sesuai dan <i>password</i> yang sesuai |
| Prosedur pengujian | 3. Penguji membuka aplikasi <i>tracer study</i> dan masuk |

| | |
|-----------------------|--|
| | kehalaman <i>login</i> |
| | 4. Penguji memasukkan <i>email</i> yang sesuai |
| | 5. Penguji memasukkan <i>password</i> yang sesuai |
| Hasil yang diharapkan | Sistem menampilkan progress bar dan menampilkan halaman utama/halaman awal |

Tabel 7.10 Kasus Uji *Register*

| | |
|-----------------------|---|
| Nomor Kasus Uji | TS-Fung-121 |
| Nama Kasus Uji | Kasus Uji <i>Register</i> |
| Kode Kebutuhan | TS-F-G-201 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan sistem dapat menampilkan halaman awal / halaman utama pada aplikasi (halaman <i>chat</i>) setelah melakukan <i>register guest</i> |
| Kasus Uji | Pengujian dilakukan dengan memasukkan <i>email</i> yang sesuai dan <i>password</i> yang sesuai |
| Prosedur pengujian | 1. Penguji membuka aplikasi <i>tracer study</i> dan masuk kehalaman <i>register</i> |
| | 2. Penguji memasukkan <i>email</i> yang sesuai |
| | 3. Penguji memasukkan kata sandi, dan konfirmasi kata sandi yang sesuai |
| | 4. Penguji memasukkan nama dan nama panggilan penguji |
| Hasil yang diharapkan | Sistem menampilkan progress bar dan menampilkan halaman utama/halaman awal |

Tabel 7.11 Kasus Uji *Register*

| | |
|------------------|--|
| Nomor Kasus Uji | TS-Fung-122 |
| Nama Kasus Uji | Kasus Uji <i>Register</i> |
| Kode Kebutuhan | TS-F-G-201 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan sistem dapat menampilkan pesan error setelah melakukan validasi <i>register guest</i> |
| Kasus Uji | Pengujian dilakukan dengan memasukkan <i>email</i> yang sesuai dan <i>password</i> yang sesuai |

| | |
|-----------------------|--|
| Prosedur pengujian | 1. Penguji membuka aplikasi <i>tracer study</i> dan masuk kehalaman <i>register</i> |
| | 2. Penguji memasukkan <i>email</i> yang sesuai |
| | 3. Penguji memasukkan kata sandi, dan konfirmasi kata sandi yang sesuai |
| Hasil yang diharapkan | Sistem menampilkan progress bar dan menampilkan pesan error karena ada data yang tidak diisi |

Tabel 7.12 Kasus Uji Kirim Pesan

| | |
|-----------------------|---|
| Nomor Kasus Uji | TS-Fung-131 |
| Nama Kasus Uji | Kasus Uji Kirim Pesan |
| Kode Kebutuhan | TS-F-AL-101 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan sistem dapat mengirimkan pesan kepada teman alumni yang sudah terdaftar didalam sistem |
| Kasus Uji | Pengujian dilakukan dengan memasukkan pesan yang ingin dikirimkan, lalu menekan tombol kirim |
| Prosedur pengujian | 1. Penguji membuka aplikasi <i>tracer study</i> dan masuk kehalaman <i>chat</i> |
| | 2. Penguji memilih teman yang akan dikirim pesan |
| | 3. Penguji memasukkan pesan yang ingin dikirimkan |
| | 4. Penguji menekan tombol kirim |
| Hasil yang diharapkan | Sistem mengirikan pesan kepada teman alumni penguji, dan menampilkan di halaman <i>chat</i> penguji |

Tabel 7.13 Kasus Uji Melihat Lowongan Pekerjaan

| | |
|--------------------|--|
| Nomor Kasus Uji | TS-Fung-141 |
| Nama Kasus Uji | Kasus Uji Melihat Lowongan Pekerjaan |
| Kode Kebutuhan | TS-F-AL-201 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan sistem dapat digunakan untuk melihat informasi lowongan pekerjaan |
| Kasus Uji | Pengujian dilakukan dengan membuka informasi pekerjaan yang sudah dimasukkan sebelumnya |
| Prosedur pengujian | 1. Penguji membuka aplikasi <i>tracer study</i> dan masuk |

| | |
|-----------------------|--|
| | kehalaman <i>chat</i> |
| | 2. Penguji membuka halaman LoKer |
| | 3. Penguji menekan salah satu lowongan yang tersedia |
| Hasil yang diharapkan | Sistem menampilkan lowongan pekerjaan sesuai dengan yang telah dimasukkan sebelumnya |

Tabel 7.14 Kasus Uji Ubah Informasi Diri

| | |
|-----------------------|--|
| Nomor Kasus Uji | TS-Fung-151 |
| Nama Kasus Uji | Kasus Uji Ubah Informasi Diri |
| Kode Kebutuhan | TS-F-AL-301 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan sistem dapat digunakan untuk mengubah informasi diri yang telah dimasukkan saat <i>register</i> |
| Kasus Uji | Pengujian dilakukan dengan mengubah data dan menyimpannya |
| Prosedur pengujian | 1. Penguji membuka aplikasi <i>tracer study</i> dan masuk kehalaman <i>chat</i> |
| | 2. Penguji membuka halaman profil |
| | 3. Penguji menekan ikon edit (bergambar pensil) |
| | 4. Penguji mengubah informasi diri, dan menyimpannya |
| Hasil yang diharapkan | Sistem menampilkan bacaan data berhasil diubah dan menampilkan halaman profil |

Tabel 7.15 Kasus Uji Masukkan Informasi Pekerjaan

| | |
|--------------------|---|
| Nomor Kasus Uji | TS-Fung-161 |
| Nama Kasus Uji | Kasus Uji Masukkkan Informasi Pekerjaan |
| Kode Kebutuhan | TS-F-AL-401 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan sistem dapat digunakan untuk memasukkan informasi pekerjaan alumni |
| Kasus Uji | Pengujian dilakukan dengan memasukkan informasi pekerjaan alumni |
| Prosedur pengujian | 1. Penguji membuka aplikasi <i>tracer study</i> dan masuk kehalaman <i>chat</i> |

| | |
|-----------------------|--|
| | 2. Penguji membuka halaman profil |
| | 3. Penguji menekan ikon tambah |
| | 4. Penguji memasukkan informasi pekerjaan dan menyimpannya |
| Hasil yang diharapkan | Sistem menampilkan pesan berhasil menyimpan dan menampilkan halaman profil |

Tabel 7.16 Kasus Uji Masukkan Informasi Pekerjaan

| | |
|-----------------------|--|
| Nomor Kasus Uji | TS-Fung-171 |
| Nama Kasus Uji | Kasus Uji Masukkkan Informasi Pendidikan |
| Kode Kebutuhan | TS-F-AL-501 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan sistem dapat digunakan untuk memasukkan informasi pendidikan alumni |
| Kasus Uji | Pengujian dilakukan dengan memasukkan informasi pendidikan alumni |
| Prosedur pengujian | 1. Penguji membuka aplikasi <i>tracer study</i> dan masuk kehalaman <i>chat</i> |
| | 2. Penguji membuka halaman profil |
| | 3. Penguji menekan ikon tambah |
| | 4. Penguji memasukkan informasi pekerjaan dan menyimpannya |
| Hasil yang diharapkan | Sistem menampilkan pesan berhasil menyimpan dan menampilkan halaman profil |

Tabel 7.17 Kasus Uji Masukkan Lowongan Pekerjaan

| | |
|--------------------|---|
| Nomor Kasus Uji | TS-Fung-181 |
| Nama Kasus Uji | Kasus Uji Masukkan Lowongan Pekerjaan |
| Kode Kebutuhan | TS-F-AL-601 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan sistem dapat digunakan untuk memasukkan lowongan pekerjaan |
| Kasus Uji | Pengujian dilakukan dengan memasukkan informasi lowongan pekerjaan tersedia |
| Prosedur pengujian | 1. Penguji membuka aplikasi <i>tracer study</i> dan masuk kehalaman <i>chat</i> |

| | |
|-----------------------|--|
| | 2. Penguji menekan tab halaman LoKer |
| | 3. Penguji menekan ikon tambah yang ada dipojok kanan aplikasi |
| | 4. Penguji memasukkan data yang dibutuhkan |
| | 5. Penguji menekan tombol simpan lowongan |
| Hasil yang diharapkan | Sistem menampilkan pesan berhasil dan berpindah ke halaman LoKer |

Tabel 7.18 Kasus Uji Melihat Informasi Pendidikan

| | |
|-----------------------|---|
| Nomor Kasus Uji | TS-Fung-191 |
| Nama Kasus Uji | Kasus Uji Melihat Informasi Pendidikan |
| Kode Kebutuhan | TS-F-AL-701 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan sistem dapat digunakan untuk melihat data pendidikan yang telah alumni isi |
| Kasus Uji | Pengujian dilakukan dengan membuka data pendidikan yang sudah dimasukkan sebelumnya |
| Prosedur pengujian | 1. Penguji membuka aplikasi <i>tracer study</i> dan masuk kehalaman <i>chat</i> |
| | 2. Penguji membuka halaman profil |
| | 3. Penguji menekan ikon lihat (bergambar mata) |
| | 4. Penguji menekan riwayat pendidikan yang sudah dimasukkan |
| Hasil yang diharapkan | Sistem menampilkan riwayat pendidikan sesuai dengan yang telah dimasukkan sebelumnya |

Tabel 7.19 Kasus Uji Melihat Informasi Pekerjaan

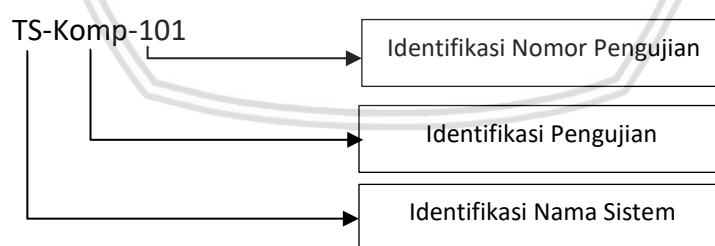
| | |
|------------------|--|
| Nomor Kasus Uji | TS-Fung-1101 |
| Nama Kasus Uji | Kasus Uji Melihat Informasi Pekerjaan |
| Kode Kebutuhan | TS-F-AL-801 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan sistem dapat digunakan untuk melihat data pekerjaan yang telah alumni isi |
| Kasus Uji | Pengujian dilakukan dengan membuka data pekerjaan yang sudah dimasukkan sebelumnya |

| | |
|-----------------------|---|
| Prosedur pengujian | 1. Penguji membuka aplikasi <i>tracer study</i> dan masuk kehalaman <i>chat</i> |
| | 2. Penguji membuka halaman profil |
| | 3. Penguji menekan ikon lihat (bergambar mata) |
| | 4. Penguji menekan riwayat pekerjaan yang sudah dimasukkan |
| Hasil yang diharapkan | Sistem menampilkan riwayat pekerjaan sesuai dengan yang telah dimasukkan sebelumnya |

7.1.2 Pengujian Kompatibilitas

Pengujian kompatibilitas dilakukan dengan memasang sistem *tracer study* pada sistem operasi yang berbeda sesuai yang telah dijelaskan dalam tabel 7.4, serta perangkat yang berbeda sesuai yang telah dijelaskan dalam tabel 7.1 hingga 7.3. Kemudian akan dilakukan beberapa kasus uji dan hasil dari pengujian tersebut akan diidentifikasi tidak valid jika berbeda dengan penggunaan pada sistem operasi yang berbeda ataupun perangkat keras yang berbeda. Namun jika hasil yang didapatkan sama pada perangkat keras yang berbeda dan sistem operasi yang berbeda, maka pengujian dikatakan valid.

Pengujian kompatibilitas akan dilakukan dengan cara mendeskripsikan kasus uji terlebih dahulu kemudian dilakukan proses pengujian, lalu akan dibandingkan kasus uji dengan hasil yang didapatkan pada pengujian. Lalu akan dibuat persentase yang akan membuktikan apakah sistem layak untuk digunakan atau tidak. Setiap kasus uji akan memiliki kode khusus dan unik yang memiliki aturan sebagai berikut :



Gambar 7.2 Aturan Penomoran Pengujian Kompatibilitas

Tabel 7.20 Kasus Uji Kompatibilitas

| Nomor Kasus Uji | Nama Kasus Uji | Hasil yang diharapkan |
|-----------------|-----------------|---|
| TS-Komp-101 | <i>Register</i> | Sistem berhasil masuk kehalaman register dengan tampilan yang sesuai dengan |

| | | |
|-------------|-------------------------------|--|
| | | implementasi yang telah dilakukan |
| TS-Komp-102 | Login | Sistem berhasil masuk kehalaman <i>login</i> dengan tampilan yang sesuai dengan implementasi yang telah dilakukan |
| TS-Komp-103 | Kirim Pesan | Sistem berhasil masuk kehalaman kirim pesan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan |
| TS-Komp-104 | Melihat Informasi Pekerjaan | Sistem berhasil masuk kehalaman lihat informasi pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan |
| TS-Komp-105 | Ubah Informasi Diri Alumni | Sistem berhasil masuk kehalaman ubah informasi diri alumni dengan tampilan yang sesuai di device yang berbeda |
| TS-Komp-106 | Masukkan Informasi Pekerjaan | Sistem berhasil masuk kehalaman masukkan informasi pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan |
| TS-Komp-107 | Masukkan Informasi Pendidikan | Sistem berhasil masuk kehalaman masukkan informasi pendidikan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan |
| TS-Komp-108 | Masukkan Lowongan Kerja | Sistem berhasil masuk kehalaman masukkan lowongan kerja dengan tampilan yang sesuai dengan implementasi yang telah dilakukan |
| TS-Komp-109 | Lihat Data Pendidikan | Sistem berhasil masuk kehalaman lihat data pendidikan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan |
| TS-Komp-110 | Lihat Data Pekerjaan | Sistem berhasil masuk kehalaman lihat data pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan |

7.1.3 Pengujian Usabilitas

Pengujian Usabilitas akan dilakukan dengan menggunakan metode *elaborated scenario based usability testing* yang mana akan menguji bagaimana setiap penguji melaksanakan hasil dari perancangan tugas pengguna dengan memberikan sedikit *clue* atau tanpa memberitahukan penguji informasi apapun tentang cara menyelesaikan tugas pengguna.

Hasil dari pengujian akan dibagi menjadi 3 bagian, yaitu berhasil, gagal, dan hampir berhasil. Berhasil adalah kondisi dimana pengguna dapat menyelesaikan semua tugas yang diberikan, sedangkan hampir berhasil adalah kondisi dimana pengujian dapat menyelesaikan sebagian tugas (lebih dari 50% tugas pengguna berhasil dilaksanakan oleh pengujian), namun gagal pada sebagian tugas lainnya, dan gagal adalah kondisi dimana pengujian gagal melakukan semua kondisi tugas pengguna. Jika berhasil maka akan dikalikan dengan 1, sedangkan jika hampir berhasil maka akan dikalikan dengan 0.5 dan jika gagal maka akan dikalikan dengan 0. Setelah dikalikan maka total nilai akan dibagi dengan total kemungkinan, jika terdapat 4 responden dengan total case 8, maka terdapat 32 total kemungkinan, setelah itu, hasil dari pembagian akan dikalikan dengan 100% (Nielsen Norman Group, 2018). Tabel 7.21 adalah contoh dari *case study* pengujian usability yang menggunakan 4 responden, sedangkan untuk pengujian nantinya akan dilakukan dengan menggunakan total 20 responden. Dari total 20 responden ini, semua adalah lulusan / alumni dari FILKOM UB yang sudah yudisium sebelum 1 Mei 2018.

Tabel 7.21 Kasus Uji Tingkat Keberhasilan Pengguna

| | Tugas Pengguna | | | | | | | |
|-----------|--|------|------|------|------|------|------|------|
| | Ke 1 | Ke 2 | Ke 3 | Ke 4 | Ke 5 | Ke 6 | Ke 7 | Ke 8 |
| Res 1 | | | | | | | | |
| Res 2 | | | | | | | | |
| Res 3 | | | | | | | | |
| Res 4 | | | | | | | | |
| Responden | Catatan : B = Berhasil; H = Hampir Berhasil; G = Gagal | | | | | | | |

Setelah itu responden juga diminta untuk mengisi kuesioner untuk penilaian sistem *tracer study*. Metode yang digunakan pada pengujian usability ini adalah pengujian dengan menggunakan metode *System Usability Testing* (SUS). Pengujian ini dilakukan dengan melibatkan target user secara langsung untuk melakukan penilaian terhadap sistem yang sedang dibangun, yang mana sistem tersebut akan diuji dengan penilaian negatif dan positif. Pertanyaan yang diberikan berjumlah 10 pertanyaan yang mana pertanyaan positif akan mengawali pertanyaan ini, dan pertanyaan negatif akan dilakukan setelah pertanyaan positif, jadi pertanyaan positif akan menjadi nomor pertanyaan ganjil, dan pertanyaan negatif akan diwakili dengan nomor pertanyaan genap. Pertanyaan akan memiliki pilihan jawaban 1 sampai 5, yang mana jawaban positif akan dikurangi satu lalu pertanyaan negatif akan diambil nilai 5 lalu dikurangi dengan jawaban yang diberikan oleh responden. Hasil akhirnya akan ditentukan dengan mengalikan jumlah total skor dengan nilai 2.5.

Berikut ini daftar pertanyaan dari kuesioner yang akan diberikan kepada responden sebagai sample pengguna dari sistem yang akan diuji dapat dilihat dalam tabel 7.22:

Tabel 7.22 Kuesioner Pengujian Usabilitas

| No | Pertanyaan | Sangat Tidak Setuju | Rating Responden | | | | Sangat Setuju |
|----|--|---------------------|------------------|---|---|---|---------------|
| | | 1 | 2 | 3 | 4 | 5 | |
| 1 | Saya pikir, saya akan sering menggunakan aplikasi ini | | | | | | |
| 2 | Saya pikir, saya menemukan kesusahan yang tidak penting dalam aplikasi ini | | | | | | |
| 3 | Menurut saya, aplikasi ini memiliki fungsi yang sangat membantu saya | | | | | | |
| 4 | Menurut saya aplikasi ini memiliki fungsi yang tidak konsisten | | | | | | |
| 5 | Saya rasa kebanyakan orang akan cepat paham dalam penggunaan aplikasi ini | | | | | | |
| 6 | Saya rasa saya membutuhkan bantuan seorang yang lebih ahli dalam penggunaan aplikasi ini | | | | | | |
| 7 | Saya merasa sangat percaya diri dalam penggunaan aplikasi ini | | | | | | |
| 8 | Saya merasa harus banyak belajar sebelum menggunakan aplikasi ini | | | | | | |
| 9 | Menurut saya aplikasi ini mudah untuk digunakan | | | | | | |
| 10 | Menurut saya aplikasi ini sangat rumit untuk digunakan | | | | | | |

Pengujian usabilitas ini akan dilakukan kepada responden-responden yang merupakan alumni dari FILKOM UB, yang telah melakukan yudisium sebelum 1 Mei 2018.

7.2 Hasil Pengujian

Dalam hasil pengujian akan disajikan hasil-hasil dari pengujian yang telah dilakukan pada perangkat keras yang telah dijabarkan sebelumnya, serta status hasil pengujian apakah valid ataupun tidak valid. Setelah hasil pengujian ini, akan dilanjutkan dengan analisis dari hasil pengujian.

7.2.1 Hasil Pengujian Fungsional

Dalam hasil pengujian fungsional terhadap sistem *tracer study* ini akan disajikan dalam bentuk tabel, lalu nantinya akan dibuatkan kesimpulan dari pengujian fungsional yang telah dilakukan. Berikut hasil pengujian fungsional yang dilakukan pada perangkat keras Samsung Galaxy S6:

Tabel 7.23 Hasil Pengujian Fungsional

| Nomor Kasus Uji | Nama Kasus Uji | Hasil yang diharapkan | Hasil Pengujian | Status |
|-----------------|--------------------|--|--|--------|
| TS-Fung-111 | Kasus Uji Login | Sistem menampilkan informasi <i>login</i> gagal, <i>email</i> tidak valid | Sistem menampilkan informasi <i>login</i> gagal, <i>email</i> tidak valid | Valid |
| TS-Fung-112 | Kasus Uji Login | Sistem menampilkan informasi <i>login</i> gagal, <i>password</i> tidak valid | Sistem menampilkan informasi <i>login</i> gagal, <i>password</i> tidak valid | Valid |
| TS-Fung-113 | Kasus Uji Login | Sistem menampilkan informasi <i>login</i> gagal, <i>email</i> harus diisi | Sistem menampilkan informasi <i>login</i> gagal, <i>email</i> harus diisi | Valid |
| TS-Fung-114 | Kasus Uji Login | Sistem menampilkan informasi <i>login</i> gagal, <i>password</i> harus diisi | Sistem menampilkan informasi <i>login</i> gagal, <i>password</i> harus diisi | Valid |
| TS-Fung-115 | Kasus Uji Login | Sistem menampilkan progress bar dan menampilkan halaman utama/halaman awal | Sistem menampilkan progress bar dan menampilkan halaman utama/halaman awal | valid |
| TS-Fung-121 | Kasus Uji Register | Sistem menampilkan progress bar dan menampilkan halaman utama/halaman awal | Sistem menampilkan progress bar dan menampilkan halaman utama/halaman awal | Valid |

| | | | | |
|--------------|--|---|---|-------|
| TS-Fung-122 | Kasus Uji <i>Register</i> | Sistem menampilkan progress bar dan menampilkan pesan error karena ada data yang tidak diisi | Sistem menampilkan progress bar dan menampilkan pesan error karena ada data yang tidak diisi | Valid |
| TS-Fung-131 | Kasus Uji Kirim Pesan | Sistem mengirim pesan kepada teman alumni penguji, dan menampilkan di halaman <i>chat</i> penguji | Sistem mengirim pesan kepada teman alumni penguji, dan menampilkan di halaman <i>chat</i> penguji | Valid |
| TS-Fung-141 | Kasus Uji Melihat Lowongan Pekerjaan | Sistem menampilkan lowongan pekerjaan sesuai dengan yang telah dimasukkan sebelumnya | Sistem menampilkan lowongan pekerjaan sesuai dengan yang telah dimasukkan sebelumnya | Valid |
| TS-Fung-151 | Kasus Uji Ubah Informasi Diri | Sistem menampilkan bacaan data berhasil diubah dan menampilkan halaman profil | Sistem menampilkan bacaan data berhasil diubah dan menampilkan halaman profil | Valid |
| TS-Fung-161 | Kasus Uji Masukkan Informasi Pekerjaan | Sistem menampilkan pesan berhasil menyimpan dan menampilkan halaman profil | Sistem menampilkan pesan berhasil menyimpan dan menampilkan halaman profil | Valid |
| TS-Fung-171 | Kasus Uji Masukkan Informasi Pendidikan | Sistem menampilkan pesan berhasil menyimpan dan menampilkan halaman profil | Sistem menampilkan pesan berhasil menyimpan dan menampilkan halaman profil | Valid |
| TS-Fung-181 | Kasus Uji Masukkan Lowongan Pekerjaan | Sistem menampilkan pesan berhasil dan berpindah ke halaman LoKer | Sistem menampilkan pesan berhasil dan berpindah ke halaman LoKer | Valid |
| TS-Fung-191 | Kasus Uji Melihat Informasi Pendidikan | Sistem menampilkan riwayat pendidikan sesuai dengan yang telah dimasukkan sebelumnya | Sistem menampilkan riwayat pendidikan sesuai dengan yang telah dimasukkan sebelumnya | Valid |
| TS-Fung-1101 | Kasus Uji Melihat Informasi | Sistem menampilkan riwayat pekerjaan sesuai dengan yang | Sistem menampilkan riwayat pekerjaan sesuai dengan yang | Valid |

| | | | | |
|--|-----------|-----------------------------|-----------------------------|--|
| | Pekerjaan | telah dimasukkan sebelumnya | telah dimasukkan sebelumnya | |
|--|-----------|-----------------------------|-----------------------------|--|

7.2.2 Hasil Pengujian Kompatibilitas

Dalam hasil pengujian ini akan diberikan hasil pengujian yang telah dilakukan terhadap perangkat keras dan perangkat lunak yang berbeda lalu akan dibuatkan kesimpulan dari pengujian kompatibilitas yang telah dilakukan. Berikut hasil pengujian kompatibilitas pada perangkat keras Samsung Galaxy S6:

Tabel 7.24 Hasil Pengujian Kompatibilitas (Samsung Galaxy S6)

| Nomor Kasus Uji | Nama Kasus Uji | Hasil yang diharapkan | Hasil Pengujian | Status |
|-----------------|-----------------------------|--|--|--------|
| TS-Komp-101 | <i>Register</i> | Sistem berhasil masuk kehalaman <i>register</i> dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman <i>register</i> dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-102 | <i>Login</i> | Sistem berhasil masuk kehalaman <i>login</i> dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman <i>login</i> dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-103 | Kirim Pesan | Sistem berhasil masuk kehalaman kirim pesan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman kirim pesan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-104 | Melihat Informasi Pekerjaan | Sistem berhasil masuk kehalaman lihat informasi pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman lihat informasi pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp- | Ubah Informasi | Sistem berhasil masuk kehalaman | Sistem berhasil masuk kehalaman ubah | valid |

| | | | | |
|--------------|-------------------------------|--|--|-------|
| 105 | Diri Alumni | ubah informasi diri alumni dengan tampilan yang sesuai di device yang berbeda | informasi diri alumni dengan tampilan yang sesuai di device yang berbeda | |
| eTS-Komp-106 | Masukkan Informasi Pekerjaan | Sistem berhasil masuk kehalaman masukkan informasi pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman masukkan informasi pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-107 | Masukkan Informasi Pendidikan | Sistem berhasil masuk kehalaman masukkan informasi pendidikan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman masukkan informasi pendidikan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-108 | Masukkan Lowongan Kerja | Sistem berhasil masuk kehalaman masukkan lowongan kerja dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman masukkan lowongan kerja dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-109 | Lihat Data Pendidikan | Sistem berhasil masuk kehalaman lihat data pendidikan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman lihat data pendidikan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-110 | Lihat Data Pekerjaan | Sistem berhasil masuk kehalaman lihat data pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman lihat data pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |

Tabel 7.25 Hasil Pengujian Kompatibilitas (Samsung J2 Pro 2016)

| Nomor Kasus Uji | Nama Kasus Uji | Hasil yang diharapkan | Hasil Pengujian | Status |
|-----------------|------------------------------|--|--|--------|
| TS-Komp-101 | <i>Register</i> | Sistem berhasil masuk kehalaman <i>register</i> dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman <i>register</i> dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-102 | <i>Login</i> | Sistem berhasil masuk kehalaman <i>login</i> dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman <i>login</i> dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-103 | Kirim Pesan | Sistem berhasil masuk kehalaman kirim pesan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman kirim pesan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-104 | Melihat Informasi Pekerjaan | Sistem berhasil masuk kehalaman lihat informasi pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman lihat informasi pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-105 | Ubah Informasi Diri Alumni | Sistem berhasil masuk kehalaman ubah informasi diri alumni dengan tampilan yang sesuai di device yang berbeda | Sistem berhasil masuk kehalaman ubah informasi diri alumni dengan tampilan yang sesuai di device yang berbeda | valid |
| eTS-Komp-106 | Masukkan Informasi Pekerjaan | Sistem berhasil masuk kehalaman masukkan informasi pekerjaan dengan tampilan yang sesuai dengan implementasi | Sistem berhasil masuk kehalaman masukkan informasi pekerjaan dengan tampilan yang sesuai dengan implementasi yang | Valid |

| | | | | |
|-------------|-------------------------------|--|--|-------|
| | | yang telah dilakukan | telah dilakukan | |
| TS-Komp-107 | Masukkan Informasi Pendidikan | Sistem berhasil masuk kehalaman masukkan informasi pendidikan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman masukkan informasi pendidikan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-108 | Masukkan Lowongan Kerja | Sistem berhasil masuk kehalaman masukkan lowongan kerja dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman masukkan lowongan kerja dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-109 | Lihat Data Pendidikan | Sistem berhasil masuk kehalaman lihat data pendidikan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman lihat data pendidikan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-110 | Lihat Data Pekerjaan | Sistem berhasil masuk kehalaman lihat data pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman lihat data pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |

Tabel 7.26 Hasil Pengujian Kompatibilitas (Emulator)

| Nomor Kasus Uji | Nama Kasus Uji | Hasil yang diharapkan | Hasil Pengujian | Status |
|-----------------|-----------------|--|--|--------|
| TS-Komp-101 | <i>Register</i> | Sistem berhasil masuk kehalaman <i>register</i> dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman <i>register</i> dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS- | <i>Login</i> | Sistem berhasil | Sistem berhasil masuk | Valid |

| | | | | |
|--------------|-------------------------------|--|--|-------|
| Komp-102 | | masuk kehalaman <i>login</i> dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | kehalaman <i>login</i> dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | |
| TS-Komp-103 | Kirim Pesan | Sistem berhasil masuk kehalaman kirim pesan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman kirim pesan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-104 | Melihat Informasi Pekerjaan | Sistem berhasil masuk kehalaman lihat informasi pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman lihat informasi pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-105 | Ubah Informasi Diri Alumni | Sistem berhasil masuk kehalaman ubah informasi diri alumni dengan tampilan yang sesuai di device yang berbeda | Sistem berhasil masuk kehalaman ubah informasi diri alumni dengan tampilan yang sesuai di device yang berbeda | valid |
| eTS-Komp-106 | Masukkan Informasi Pekerjaan | Sistem berhasil masuk kehalaman masukkan informasi pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman masukkan informasi pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-107 | Masukkan Informasi Pendidikan | Sistem berhasil masuk kehalaman masukkan informasi pendidikan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman masukkan informasi pendidikan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-108 | Masukkan Lowongan Kerja | Sistem berhasil masuk kehalaman masukkan lowongan kerja dengan | Sistem berhasil masuk kehalaman masukkan lowongan kerja dengan tampilan yang | Valid |

| | | | | |
|-------------|-----------------------|--|--|-------|
| | | tampilan yang sesuai dengan implementasi yang telah dilakukan | sesuai dengan implementasi yang telah dilakukan | |
| TS-Komp-109 | Lihat Data Pendidikan | Sistem berhasil masuk kehalaman lihat data pendidikan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman lihat data pendidikan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |
| TS-Komp-110 | Lihat Data Pekerjaan | Sistem berhasil masuk kehalaman lihat data pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Sistem berhasil masuk kehalaman lihat data pekerjaan dengan tampilan yang sesuai dengan implementasi yang telah dilakukan | Valid |

7.2.3 Hasil Pengujian Usabilitas

Tabel 7.27 Hasil Pengujian Usabilitas

| | Tugas Pengguna | | | | | | | |
|--------|----------------|------|------|------|------|------|------|------|
| | Ke 1 | Ke 2 | Ke 3 | Ke 4 | Ke 5 | Ke 6 | Ke 7 | Ke 8 |
| Res 1 | B | H | B | B | B | B | B | B |
| Res 2 | B | H | G | H | B | B | B | B |
| Res 3 | B | H | H | G | H | G | H | B |
| Res 4 | B | H | B | B | B | B | B | B |
| Res 5 | B | B | B | B | B | B | B | B |
| Res 6 | B | B | B | B | B | B | B | B |
| Res 7 | B | B | H | H | H | H | G | B |
| Res 8 | B | B | B | B | B | H | H | B |
| Res 9 | B | B | B | B | B | B | B | B |
| Res 10 | B | B | B | B | B | B | B | B |
| Res 11 | B | B | B | B | B | B | B | B |
| Res 12 | B | B | B | B | B | B | B | B |
| Res 13 | B | B | B | B | B | B | B | B |

| | | | | | | | | |
|-----------|--|---|---|---|---|---|---|---|
| Res 14 | B | B | B | B | B | B | B | B |
| Res 15 | B | B | B | B | B | B | B | B |
| Res 16 | B | B | B | B | B | B | B | B |
| Res 17 | B | B | B | B | B | B | B | B |
| Res 18 | B | B | B | B | B | B | B | B |
| Res 19 | B | B | B | H | B | H | B | B |
| Res 20 | B | B | G | G | B | B | B | B |
| Responden | Catatan : B = Berhasil; H = Hampir Berhasil; G = Gagal | | | | | | | |

Res (responden) : Alumni FILKOM yang sudah yudisium sebelum 1 Mei 2018

$$Success\ rate = \frac{(B + (H \times 0.5))}{jumlah\ tugas \times jumlah\ responden} \times 100\% \quad (7.1)$$

$$Success\ rate = \frac{(138 + (16 \times 0.5))}{8 \times 20} \times 100\% = 91.25\% \quad (7.2)$$

Dari total 160 percobaan dengan total 20 responden (Dari total 20 responden ini, semua adalah lulusan / alumni dari FILKOM UB yang sudah yudisium sebelum 1 Mei 2018.), maka didapatkan sebanyak total 138 kali responden berhasil menyelesaikan tugas pengguna yang diminta. Sedangkan 16 kali kesempatan, responden hampir berhasil melakukannya tanpa masalah, sedangkan 6 kali kesempatan, responden gagal menyelesaikan tugas pengguna yang diminta. Setelah dilakukan proses uji ini, maka akan dilakukan proses perhitungan dari hasil pengujian yang telah dilakukan.

Dengan total nilai hasil pengujian *success rate* sebesar 91.25%, responden dapat dianggap berhasil menggunakan aplikasi tanpa masalah yang berarti, sehingga dapat disimpulkan responden tidak kesulitan dalam penggunaan aplikasi *tracer study*.

Sedangkan hasil pengujian dengan menggunakan metode *System Usability Testing* (SUS) akan dilampirkan dalam lampiran D. dari hasil tersebut, dapat dibuat total dari hasil yang diisi oleh responden.

Tabel 7.28 Hasil Pengujian Usabilitas

| No | Tahun Angkatan Responden | Total Skor |
|----|--------------------------|------------|
| 1 | 2014 | 28 |
| 2 | 2013 | 29 |
| 3 | 2014 | 33 |
| 4 | 2014 | 27 |
| 5 | 2014 | 28 |

| | | |
|------------|------|-----|
| 6 | 2014 | 34 |
| 7 | 2013 | 34 |
| 8 | 2013 | 40 |
| 9 | 2013 | 34 |
| 10 | 2013 | 29 |
| 11 | 2014 | 28 |
| 12 | 2013 | 38 |
| 13 | 2010 | 32 |
| 14 | 2014 | 28 |
| 15 | 2010 | 27 |
| 16 | 2013 | 34 |
| 17 | 2013 | 35 |
| 18 | 2014 | 35 |
| 19 | 2012 | 36 |
| 20 | 2013 | 36 |
| Total Skor | | 645 |

$$Usability = \frac{\sum Total\ skor}{\sum Responden} \times 2.5 \quad (7.3)$$

$$Usability = \frac{645}{20} \times 2.5 = 80.625 \quad (7.4)$$

7.3 Analisis Hasil Pengujian

Dari hasil pengujian yang telah dilakukan sebelumnya, akan dilakukan analisis terhadap masing masing hasil pengujian, yang mana nantinya akan menghasilkan sebuah kesimpulan apakah sistem berhasil melewati pengujian yang telah dilakukan ataupun sistem telah gagal melewati pengujian yang telah dilakukan.

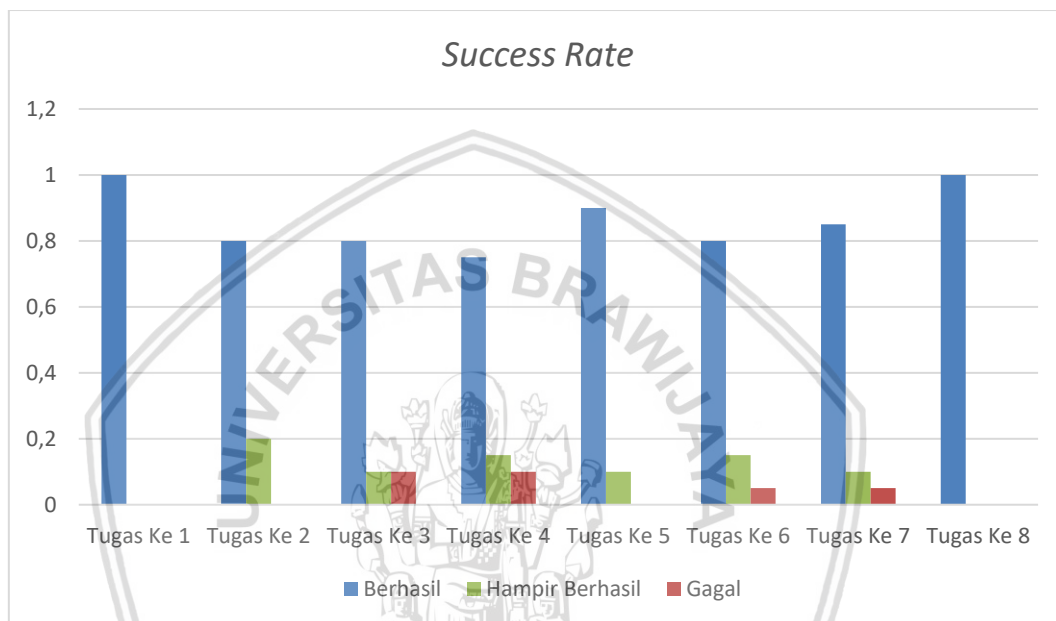
7.3.1 Analisis Hasil Pengujian Fungsional

Berdasarkan hasil pengujian fungsional yang telah dilakukan sebelumnya, sistem dapat melakukan semua fungsional sistem dengan baik dan tanpa masalah. Dapat disimpulkan bahwa sistem memiliki fungsionalitas yang baik untuk perangkat yang telah diuji yaitu dengan tingkat keberhasilan 100%. Diharapkan sistem memiliki tingkat fungsionalitas yang baik pada perangkat lain. Untuk memastikan sistem dapat berjalan diperangkat lain maka telah dilakukan uji kompatibilitas sebelumnya, hasilnya dapat dilihat di sub bab 7.3.2.

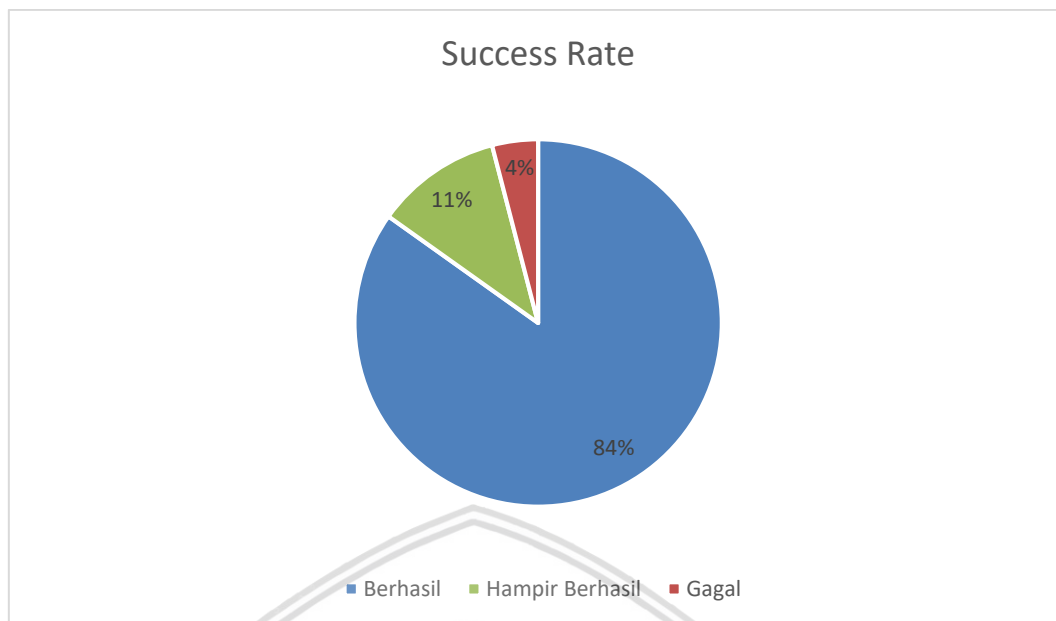
7.3.2 Analisis Hasil Pengujian Kompatibilitas

Berdasarkan hasil pengujian kompatibilitas yang telah dilakukan sebelumnya, sistem dapat melewati semua pengujian dengan baik dan tanpa error. Dapat disimpulkan bahwa sistem memiliki tingkat usabilitas yang cukup baik untuk 3 jenis perangkat keras yang berbeda dan 2 sistem operasi yang berbeda. Diharapkan sistem memiliki tingkat kompatibilitas yang baik pada perangkat lunak dan perangkat keras yang sejenis.

7.3.3 Analisis Hasil Pengujian Usabilitas



Berdasarkan hasil pengujian success rate pada pengujian usabilitas, didapatkan bahwa pada tugas pengguna ke 1 dan ke 8, semua pengguna berhasil menyelesaikan tugas tersebut, sedangkan untuk tugas pengguna ke 4 adalah tugas pengguna yang dirasa susah oleh pengguna dengan total 15 kali berhasil.



Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa responden alumni dapat menggunakan aplikasi dengan mudah tanpa kendala. Dengan skor hasil pengujian usabilitas yang menunjang pertanyaan ini yaitu total rata-rata hasil pengujian adalah 80.625 yang didapatkan pada sampel responden sudah melebihi standar skor kepuasan pengguna yaitu 68. Standar kepuasan yang baik pada penilaian SUS adalah pada skor 68 (Sauro, 2011).

BAB 8 PENUTUP

8.1 Kesimpulan

Berdasarkan hasil analisis kebutuhan, perancangan, implementasi, dan pengujian yang telah dilakukan sebelumnya oleh, maka dapat diambil kesimpulan sebagai berikut:

1. Analisis kebutuhan yang didapatkan dalam proses wawancara dengan alumni, koordinator alumni, dosen serta bagian kemahasiswaan yang mengurus *tracer study* menghasilkan 10 kebutuhan fungsional serta 2 kebutuhan non fungsional yang dapat diaplikasikan menjadi aplikasi berbasis Android.
2. Rancangan sistem *tracer study* berbentuk sebuah aplikasi *mobile* berbasis Android yang didalamnya terdapat fungsi fungsi untuk saling berkirim pesan, melihat lowongan pekerjaan, mengisikan riwayat pendidikan, dan riwayat pekerjaan bagi alumni. Dimana perancangannya telah dilakukan pada bab perancangan yaitu dengan merancang diagram activity, merancang data, merancang antarmuka, serta merancang tugas pengguna.
3. Implementasi sistem *tracer study* berbentuk aplikasi *mobile* berbasis Android yang memanfaatkan *platform* firebase dapat dilihat pada bagian implementasi kode program, dimana aplikasi memanfaatkan firebase *realtime database*, firebase *authenticate*, firebase *storage* untuk membantu dalam proses implementasi aplikasi *tracer study* dapat berjalan dengan baik.
4. Tingkat usability pada aplikasi *tracer study* ini dapat dikatakan baik karena memiliki hasil 80,625% dimana sudah melebihi standar nilai usability yaitu 68%. Serta memiliki tingkat kompatibilitas yang baik karena memiliki hasil 100% berhasil dari 3 perangkat keras berbeda serta 2 sistem operasi berbeda.

8.2 Saran

Adapun saran untuk pengembangan sistem selanjutnya adalah sebagai berikut:

1. Fitur untuk mencari lowongan pekerjaan yang ada dihalaman lowongan pekerjaan.
2. Perbaiki antarmuka dan user experience untuk kepuasan pengguna yang lebih baik.
3. Penggunaan *database* dari filkom untuk *login* dan mengambil serta menyimpan data alumni sehingga alumni tidak perlu mengisi banyak data, dan sistem dapat memvalidasi alumni benar benar lulusan filkom.

4. Sistem bisa dikembangkan lebih jauh untuk mengolah data dari tracer study



DAFTAR PUSTAKA

OpenUP, 2008. *Conceito: Entity-Control-Boundary pattern*. [Online]
Available at:
http://epf.eclipse.org/wikis/openuppt/openup_basic/guidances/concepts/entity_control_boundary_pattern, uF-QYEAhEdg UJTvM1DM2Q.html
[Accessed 7 Maret 2018].

Aditya Orba Ramadhan, H. T. L. F., 2017. Pembangunan Modul Penunjang Pembelajaran di Kelas untuk Aplikasi Brawijaya Messenger dengan Platform Firebase. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 1(1), pp. 1-7.

Bohang, F. K., 2016. *Evolusi OS Android, dari Versi 1.0 hingga 7.0 Nougat*. [Online]
Available at:
<https://tekno.kompas.com/read/2016/10/17/19480037/evolusi.os.android.dari.versi.1.0.hingga.7.0.nougat?page=all>
[Accessed 7 Maret 2018].

DIKTI, n.d. *Tracer Studi - About Site*. [Online]
Available at: <http://tracerstudy.dikti.go.id/index.php/site/about>
[Accessed 5 February 2018].

Firebase, n.d. *Firebase*. [Online]
Available at: <https://firebase.google.com/>
[Accessed 31 01 2018].

Isparmo, 2016. *Data Statistik Pengguna Internet Indonesia 2016*. [Online]
Available at: <http://isparmo.web.id/2016/11/21/data-statistik-pengguna-internet-indonesia-2016/>
[Accessed 7 Maret 2018].

Job Placement Center dan Tracer Study Universitas Brawijaya, 2018. <http://tracer-study.ub.ac.id/#tentang>. [Online]
Available at: <http://www.tracer-study.ub.ac.id>
[Accessed 23 January 2018].

Json, n.d. *JSON*. [Online]
Available at: <https://www.json.org/>
[Accessed 6 February 2018].

Kompas, 2017. *Inilah Jumlah Pengguna Internet Pakai Smartphone dan Aplikasi Mobile*. [Online]
Available at: <http://jogja.tribunnews.com/2017/04/01/inilah-jumlah-pengguna-internet-pakai-smartphone-dan-aplikasi-mobile-terpopuler-di-indonesia>
[Accessed 8 February 2018].

Kumar, K., Akhi, K., Gunti, S. K. & Reddy, M. P., 2016. Implementing Smart Home Using Firebase. *International Journal of Research in Engineering and Applied Sciences (IJREAS)*, 6(10), pp. 193-198.

Mfc, S., 2015. *Pentingnya Tracer Study untuk Akreditasi dan Evaluasi Kampus*. [Online]

Available at: <https://studentstelkomuniversity.com/pentingnya-tracer-study-untuk-akreditasi-dan-evaluasi-kampus/>

[Accessed 1 February 2018].

Nielsel Norman Group, 2018. *Success Rate: The Simplest Usability Metric*. [Online]

Available at: <https://www.nngroup.com/articles/success-rate-the-simplest-usability-metric/>

[Accessed 27 Mei 2018].

Nielsen Norman Group, 2018. *Usability 101: Introduction to Usability*. [Online]

Available at: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>

[Accessed 13 March 2018].

Pressman, R. S., 2008. *Software Engineering-A Practitioners Approach - 7th Edition*. 7th ed. New York: McGraw-Hill.

Rachman, A. F., 2015. *Android Kuasai Asia Tenggara*. [Online]

Available at: <https://inet.detik.com/consumer/d-3054169/android-kuasai-asia-tenggara-di-indonesia-paling-juara>

[Diakses 24 January 2018].

Rozak, A., Tolle, H. & Fanani, L., 2018. Pembangunan Aplikasi Brawijaya Messenger dengan menggunakan Platform Firebase pada Universitas Brawijaya. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(2), pp. 667-673.

Sauro, J., 2011. *Measuring Usability with the System Usability Scale (SUS)*. s.l.:s.n.